

ロボティクス Pick & Place ソリューション

技術導入ガイド ビジョン&ロボット統合シミュレーション キャリブレーションパラメータ編

形 SYSMAC-SE20□□

形 SYSMAC-RA401L

形 NJ501-4□□□

形 R88D-KN□-ECT

形 FH-1□□□

形 FH-3□□□

Technology
Introduction
Guide

— おことわり —

- (1) 本ガイドの内容の一部または全部を無断で複写、複製、転載することを禁じます。
- (2) 本ガイドの内容に関しては、改良のため予告なしに仕様などを変更することがあります。
あらかじめご了承ください。
- (3) 本ガイドの内容に関しては、万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがありましたら、お手数ですが弊社支店または営業所までご連絡ください。その際、巻末記載のMan.No(マニュアルNo.)も併せてお知らせください。

はじめに

技術導入ガイド ビジョン&ロボット統合シミュレーション キャリブレーションパラメータ編(以降「本ガイド」と省略する場合があります。)は、Sysmac Studio のシミュレーション機能の一つである、ビジョン&ロボット統合シミュレーションが算出する、キャリブレーションパラメータをご説明する資料です。

なお、本ガイドには安全に関する注意事項を含め、使用する際の詳細内容については記載していません。必ず、本ガイドで使用する各機器のマニュアルや取扱説明書を入手し、「安全上のご注意」「安全上の要点」「使用上の注意」などの安全に関する注意事項を含め、使用する際の詳細内容を確認したうえで使用してください。

ビジョン&ロボット統合シミュレーションを使用する基本的な手順と操作については、□『ビジョン&ロボット統合シミュレーション スタートアップガイド(SAMC-002)』を参照してください。

対象となる読者の方々

本ガイドは、次の方を対象に記述しています。

電気の知識(電気工事士あるいは同等の知識)を有する方で

- ・FA 機器の導入を担当される方
- ・FA システムを設計される方
- ・FA 現場を管理される方

また、プログラミング言語に関して、国際標準規格 IEC 61131-3 もしくは国内標準規格 JIS B 3503 に規定される内容をご理解いただいている方を対象に記述しています。

対象となる製品

本ガイドは、以下の製品を対象にしています。

- ・マシンオートメーションコントローラ NJ シリーズ CPU ユニット
- ・オートメーションソフトウェア Sysmac Studio
- ・AC サーボモータドライバ G5 シリーズ
- ・画像センサ FH シリーズ 画像処理システム
- ・プログラマブルターミナル NA シリーズ

ご承諾事項

「当社商品」について特別の合意がない場合には、お客様のご購入先にかかわらず、本ご承諾事項記載の条件を適用いたします。

● 定義

本ご承諾事項中の用語の定義は次のとおりです。

- ・「当社商品」:「当社」の FA システム機器、汎用制御機器、センシング機器、電子・機構部品
- ・「カタログ等」:「当社商品」に関する、ベスト制御機器オムロン、電子・機構部品総合カタログ、その他のカタログ、仕様書、取扱説明書、マニュアル等であって電磁的方法で提供されるものも含まれます。
- ・「利用条件等」:「カタログ等」に記載の、「当社商品」の利用条件、定格、性能、動作環境、取り扱い方法、利用上の注意、禁止事項その他
- ・「お客様用途」:「当社商品」のお客様におけるご利用方法であって、お客様が製造する部品、電子基板、機器、設備またはシステム等への「当社商品」の組み込み、または利用を含みます。
- ・「適合性等」:「お客様用途」での「当社商品」の(a)適合性、(b)動作、(c)第三者の知的財産の非侵害、(d)法令の遵守および(e)各種規格の遵守

● 記載事項のご注意

「カタログ等」の記載内容については次の点をご理解ください。

- ・定格値および性能値は、単独試験における各条件のもとで得られた値であり、各定格値および性能値の複合条件のもとで得られる値を保証するものではありません。
- ・参考データはご参考として提供するもので、その範囲で常に正常に動作することを保証するものではありません。
- ・利用事例はご参考ですので、「当社」は「適合性等」について保証いたしかねます。
- ・「当社」は、改善や当社都合等により、「当社商品」の生産を中止し、または「当社商品」の仕様を変更することがあります。

● ご利用にあたってのご注意

ご採用およびご利用に際しては次の点をご理解ください。

- ・定格・性能ほか「利用条件等」を遵守しご利用ください。
- ・お客様ご自身にて「適合性等」をご確認いただき、「当社商品」のご利用の可否をご判断ください。「当社」は「適合性等」を一切保証いたしかねます。
- ・「当社商品」がお客様のシステム全体の中で意図した用途に対して、適切に配電・設置されていることをお客様ご自身で、必ず事前に確認してください。
- ・「当社商品」をご使用の際には、(i)定格および性能に対し余裕のある「当社商品」のご利用、冗長設計などの安全設計、(ii)「当社商品」が故障しても、「お客様用途」の危険を最小にする安全設計、(iii)利用者に危険を知らせるための、安全対策のシステム全体としての構築、(iv)「当社商品」および「お客様用途」の定期的な保守、の各事項を実施してください。
- ・「当社」は DDoS 攻撃(分散型 DoS 攻撃)、コンピュータウイルスその他の技術的な有害プログラム、不正アクセスにより、「当社商品」、インストールされたソフトウェア、またはすべてのコンピュータ機器、コンピュータプログラム、ネットワーク、データベースが感染したとしても、そのことにより直接または間接的に生じた損失、損害その他の費用について一切責任を負わないものとします。
お客様ご自身にて、①アンチウイルス保護、②データ入出力、③紛失データの復元、④「当社商品」またはインストールされたソフトウェアに対するコンピュータウイルス感染防止、⑤「当社商品」に対する不正アクセス防止についての十分な措置を講じてください。
- ・「当社商品」は、一般工業製品向けの汎用品として設計製造されています。従いまして、次に掲げる用途での

使用は意図しておらず、お客様が「当社商品」をこれらの用途に使用される際には、「当社」は「当社商品」に対して一切保証をいたしません。ただし、次に掲げる用途であっても

「当社」の意図した特別な商品用途の場合や特別の合意がある場合は除きます。

- (a) 高い安全性が必要とされる用途(例: 原子力制御設備、燃焼設備、航空・宇宙設備、鉄道設備、昇降設備、娯楽設備、医用機器、安全装置、その他生命・身体に危険が及びうる用途)
 - (b) 高い信頼性が必要な用途(例: ガス・水道・電気等の供給システム、24 時間連続運転システム、決済システムほか権利・財産を取扱う用途など)
 - (c) 厳しい条件または環境での用途(例: 屋外に設置する設備、化学的汚染を被る設備、電磁的妨害を被る設備、振動・衝撃を受ける設備など)
 - (d) 「カタログ等」に記載のない条件や環境での用途
- ・上記の(a)から(d)に記載されている他、「本カタログ等記載の商品」は自動車(二輪車含む。以下同じ)向けではありません。自動車に搭載する用途には利用しないで下さい。自動車搭載用商品については当社営業担当者にご相談ください。

● 保証条件

「当社商品」の保証条件は次のとおりです。

・保証期間 ご購入後 1 年間といたします。

(ただし「カタログ等」に別途記載がある場合を除きます。)

・保証内容 故障した「当社商品」について、以下のいずれかを「当社」の任意の判断で実施します。

(a) 当社保守サービス拠点における故障した「当社商品」の無償修理

(ただし、電子・機構部品については、修理対応は行いません。)

(b) 故障した「当社商品」と同数の代替品の無償提供

・保証対象外 故障の原因が次のいずれかに該当する場合は、保証いたしません。

(a) 「当社商品」本来の使い方以外のご利用

(b) 「利用条件等」から外れたご利用

(c) 本ご承諾事項「ご利用にあたってのご注意」に反するご利用

(d) 「当社」以外による改造、修理による場合

(e) 「当社」以外の者によるソフトウェアプログラムによる場合

(f) 「当社」からの出荷時の科学・技術の水準では予見できなかった原因

(g) 上記のほか「当社」または「当社商品」以外の原因(天災等の不可抗力を含む)

● 責任の制限

本ご承諾事項に記載の保証が、「当社商品」に関する保証のすべてです。

「当社商品」に関連して生じた損害について、「当社」および「当社商品」の販売店は責任を負いません。

● 輸出管理

「当社商品」または技術資料を、輸出または非居住者に提供する場合は、安全保障貿易管理に関する日本および関係各国の法令・規制を遵守ください。お客様が法令・規則に違反する場合には、「当社商品」または技術資料をご提供できない場合があります。

注意事項

- ・実際のシステム構築に際しては、システムを構成する各機器や装置の仕様をご確認のうえ、定格や性能に対し余裕を持った使い方をしてください。万一故障があっても危険を最小にする安全回路などの安全対策を講じてください。
- ・システムを安全に使用していただくため、システムを構成する各機器や装置のマニュアルや取扱説明書などを入手してください。「安全上のご注意」「安全上の要点」「使用上の注意」など安全に関する注意事項を含め、内容を確認のうえ使用してください。
- ・システムが適合すべき規格や法規、または規制については、お客様自身で確認してください。

商標

- ・Sysmac は、オムロン株式会社製 FA 機器製品の日本およびその他の国における商標または登録商標です。
 - ・Windows、Windows 98、Windows XP、Windows Vista、Windows 7 は米国 Microsoft Corporation の米国およびその他の国における登録商標です。
 - ・Intel、インテル、Intel ロゴ、インテル Atom は、米国およびその他の国におけるインテル コーポレーションの商標です。
 - ・EtherCAT® は、ドイツ Beckhoff Automation GmbH によりライセンスされた特許取得済み技術であり登録商標です。
 - ・SD、SDHC ロゴは、SD-3C,LLC の商標です。
- その他、記載されている会社名と製品名などにつきましては、各社の登録商標または商標です。

ソフトウェアのライセンスと著作権

本製品にはサードパーティ製のソフトウェアが組み込まれています。このソフトウェアに関連するライセンスと著作権については、http://www.fa.omron.co.jp/nj_info_j/ をご覧ください。

マニュアル改訂履歴

マニュアル改訂記号は、表表紙・裏表紙に記載されている Man.No.の語尾に付記されます。

Man.No. SAMC-008B

改訂記号

改訂記号	改訂年月	改訂理由・改訂ページ
A	2015 年 12 月	初版
B	2018 年 12 月	ご承諾事項の変更

目次

はじめに	1
ご承諾事項.....	2
注意事項.....	4
マニュアル改訂履歴.....	5

第 1 章 概要

1-1 概要	1-2
1-2 キャリブレーションパラメータ	1-3

第 2 章 変数とデータ型

2-1 変数	2-2
2-2 データ型(構造体型)	2-3

第 3 章 パラメータの説明

3-1 ロボットに関するデータ	3-2
3-2 搬入コンベアに関するデータ	3-10
3-3 搬出コンベアに関するデータ	3-14
3-4 画像センサに関するデータ.....	3-17
3-4-1 画像センサの通信コマンドマクロ例	3-19
3-4-2 CPU ユニットのプログラム例	3-25

第 4 章 座標系の説明

4-1 座標系の基本.....	4-2
4-2 統合シミュレーションの座標系	4-3
4-2-1 搬入コンベアの座標系	4-3
4-2-2 搬入コンベアとロボットの関係	4-3
4-2-3 搬入コンベアと画像センサのカメラ視野の関係	4-7
4-2-4 搬出コンベアの座標系	4-9
4-2-5 搬出コンベアの ON/OFF センサについて.....	4-9
4-2-6 搬出コンベアとロボットの関係	4-10

1

概要

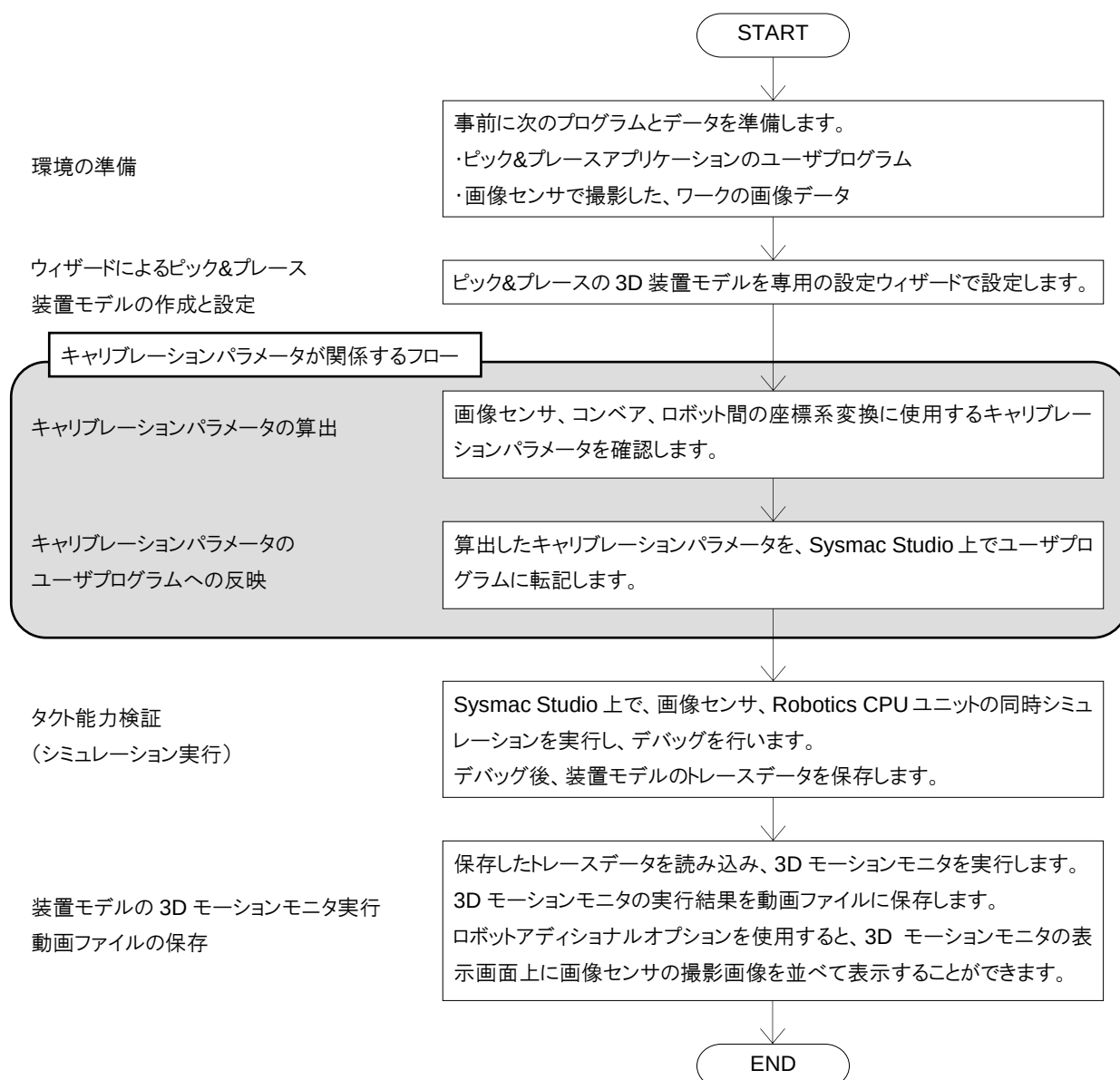
この章では、ビジョン&ロボット統合シミュレーションのキャリブレーションパラメータの概要について説明しています。

1-1	概要	1-2
1-2	キャリブレーションパラメータ	1-3

1-1 概要

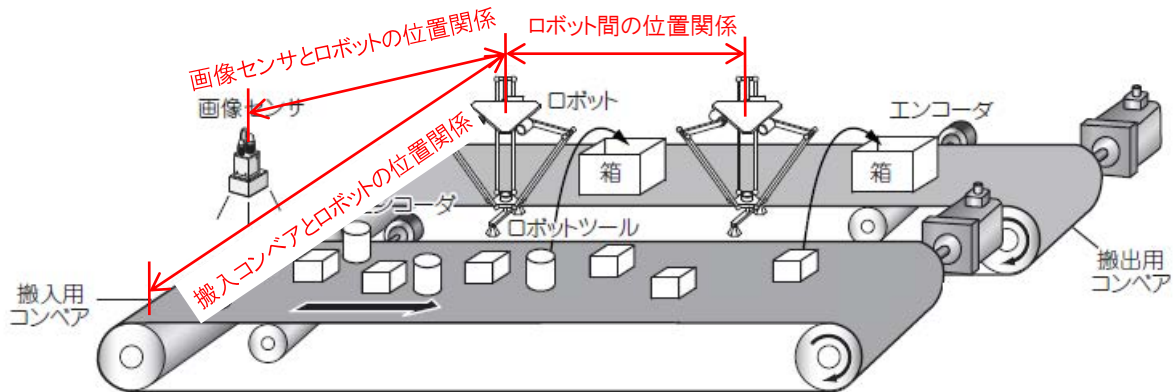
本ガイドは、Sysmac Studio のシミュレーション機能の一つである、ビジョン&ロボット統合シミュレーション（以降「統合シミュレーション」と省略します。）が算出する、キャリブレーションパラメータをご説明する資料です。

統合シミュレーションのフローにおける、「キャリブレーションパラメータの算出」および「キャリブレーションパラメータのユーザプログラムへの反映」にて、使用するための資料です。



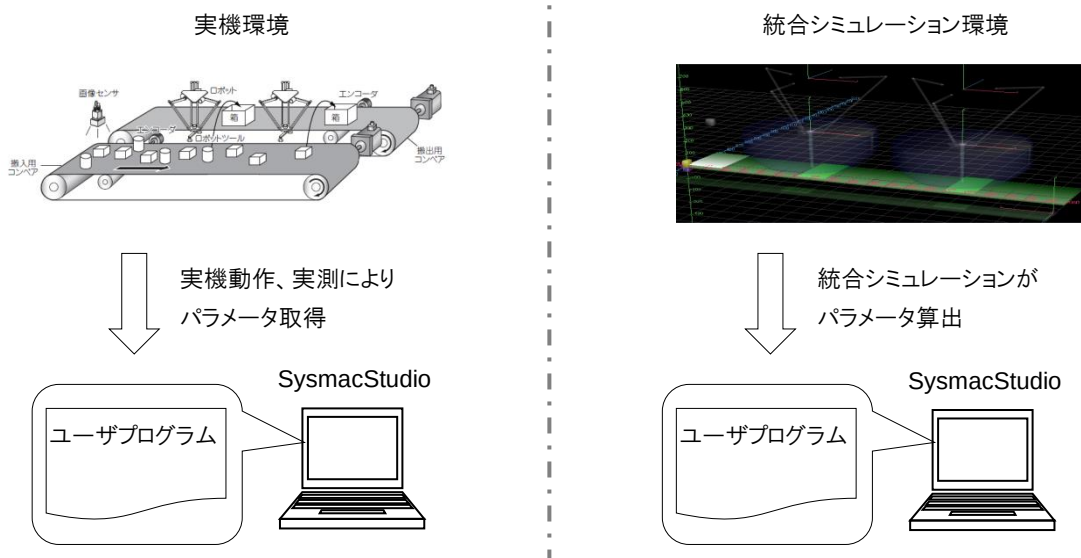
1-2 キャリブレーションパラメータ

キャリブレーションパラメータとは、ピック&プレースの3D 装置モデルにおける、画像センサ、コンベア、ロボット、等の各 3D 機構モデル間の位置関係や向き、各 3D 機構モデルのパラメータ設定、等の各種データです。



実機環境では、実機の画像センサ、コンベア、ロボットを動作させて、実測することにより、これらのキャリブレーションパラメータを得ることが出来ますが、統合シミュレーションでは、専用の設定ウィザードを使用して、3D 装置モデルを設定した時に、これらのパラメータが算出されます。

キャリブレーションパラメータを、ピック&プレースアプリケーションのユーザプログラムに適切に反映させることで、ピック&プレースアプリケーションの動作を 3D でシミュレーションすることが可能になります。



キャリブレーションパラメータデータの出力

統合シミュレーションを使用して、3D 装置モデルを設定した後に、専用の設定ウィザードで[パラメータ出力(ST プログラム)]を選択するか、3D モーションモニタビューで[キャリブレーションパラメータ出力]を選択すれば、3D 装置モデルのキャリブレーションパラメータがテキストファイル(*.txt)に保存されます。

設定ウィザードで設定完了時に

パラメータ出力(STプログラム)...

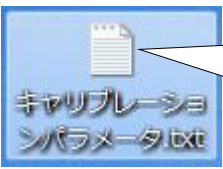
をクリックする

または

3D モーションモニタビューで



をクリックする



キャリブレーションパラメータが出力されたテキストファイルが保存されます。

```
SimData.PickLine[1].ConfigTable[0].RobotNumber:=1;
SimData.PickLine[1].ConfigTable[0].CSID:=2;
SimData.PickLine[1].ConfigTable[0].SensorType:=1;
SimData.PickLine[1].ConfigTable[0].CameraSensorNumber:=0;

SimData.PickLine[1].ConfigTable[1].RobotNumber:=2;
SimData.PickLine[1].ConfigTable[1].CSID:=2;

SimData.PlaceLine[1].ConfigTable[0].RobotNumber:=2;
SimData.PlaceLine[1].ConfigTable[0].CSID:=3;
SimData.PlaceLine[1].ConfigTable[0].SensorType:=2;

SimData.PlaceLine[1].ConfigTable[1].RobotNumber:=1;
SimData.PlaceLine[1].ConfigTable[1].CSID:=3;

SimData.Robot[1].UCSoverMCS1[2,0]:=-722;
SimData.Robot[1].UCSoverMCS1[2,1]:=25;
SimData.Robot[1].UCSoverMCS1[2,2]:=-770;
SimData.Robot[1].UCSoverMCS1[2,3]:=0;
SimData.Robot[1].UCSoverMCS1[2,4]:=0;
SimData.Robot[1].UCSoverMCS1[2,5]:=0;
```

キャリブレーションパラメータデータの使用方法

テキストファイルに出力されたキャリブレーションパラメータを使用するために、デバッグプログラム(ST プログラム)を作成し、テキストファイルの内容を転記してください。

マルチビューエクスプローラ

new_Controller_0

- 構成・設定
- プログラミング
 - POU
 - プログラム
 - Program0
- データ
 - データ型
 - グローバル変数
- タスク

Program0

変数テーブル

```
1 SimData.PickLine[1].ConfigTable[0].RobotNumber:=1;
2 SimData.PickLine[1].ConfigTable[0].CSID:=2;
3 SimData.PickLine[1].ConfigTable[0].SensorType:=1;
4 SimData.PickLine[1].ConfigTable[0].CameraSensorNumber:=0;
5
6 SimData.PickLine[1].ConfigTable[1].RobotNumber:=2;
7 SimData.PickLine[1].ConfigTable[1].CSID:=2;
8
9 SimData.PlaceLine[1].ConfigTable[0].RobotNumber:=1;
10 SimData.PlaceLine[1].ConfigTable[0].CSID:=3;
11 SimData.PlaceLine[1].ConfigTable[0].SensorType:=2;
12
13 SimData.PlaceLine[1].ConfigTable[1].RobotNumber:=2;
14 SimData.PlaceLine[1].ConfigTable[1].CSID:=3;
15
16 SimData.Robot[1].UCSoverMCS1[2,0]:=-1100;
17 SimData.Robot[1].UCSoverMCS1[2,1]:=-400;
18 SimData.Robot[1].UCSoverMCS1[2,2]:=-950;
19 SimData.Robot[1].UCSoverMCS1[2,3]:=0;
20 SimData.Robot[1].UCSoverMCS1[2,4]:=0;
21 SimData.Robot[1].UCSoverMCS1[2,5]:=0;
22
23 SimData.Robot[1].UCS[2].CSID:=2;
24 SimData.Robot[1].UCS[2].Pose[0]:=-1100;
25 SimData.Robot[1].UCS[2].Pose[1]:=-400;
26 SimData.Robot[1].UCS[2].Pose[2]:=-950;
27 SimData.Robot[1].UCS[2].Pose[3]:=0;
28 SimData.Robot[1].UCS[2].Pose[4]:=0;
29 SimData.Robot[1].UCS[2].Pose[5]:=0;
30
```

編集(E)

切り取り(T)

コピー(C)

貼り付け(P)

削除(D)

名前の変更(R)

上に挿入

上に移動

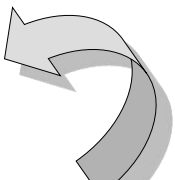
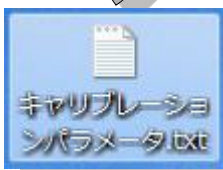
下に移動

セキュリティ(U)

プロパティ

デバッグ用設定

有効

2

変数とデータ型

この章では、ビジョン&ロボット統合シミュレーションが算出したキャリブレーションパラメータを代入するための変数、および、変数のデータ型について説明しています。

2-1	変数	2-2
2-2	データ型(構造体型)	2-3

2-1 変数

統合シミュレーションが算出したキャリブレーションパラメータは、変数にデータを代入する ST プログラムとして、テキストファイル(*.txt)に出力されます。

算出したキャリブレーションパラメータの ST プログラムを、ユーザプログラムで使用するために、以下の変数をローカル変数、または、グローバル変数に定義してください。

変数名	データ型	用途
SimData	sSimData	ビジョン&ロボット統合シミュレーションが算出したキャリブレーションパラメータを代入する

2-2 データ型(構造体型)

キャリブレーションパラメータを代入する変数は、データ型が構造体型です。
また、構造体型変数のメンバに、データ型が構造体型のメンバがあります。

キャリブレーションパラメータを代入する変数を使用するために、以下の構造体変数を定義してください。

- ・ sSimData
- ・ sSimRobot
- ・ sSimOpeRange
- ・ sSimLine
- ・ sLineConfigTable
- ・ sSimVision

sSimData の仕様

構造体型 sSimData の仕様は、以下のとおりです

名称	データ型	内容
sSimData	STRUCT	
Robot	ARRAY[0..8] OF sSimRobot	ロボットに関連するデータ群
PickLine	ARRAY[0..16] OF sSimLine	搬入コンベアに関するデータ群
PlaceLine	ARRAY[0..16] OF sSimLine	搬出コンベアに関するデータ群
Vision	ARRAY[0..7] OF sSimVision	画像センサに関連するデータ群

sSimRobot の仕様

構造体型 sSimRobot の仕様は、以下のとおりです

名称	データ型	内容
sSimRobot	STRUCT	
UCSoverMCS1	ARRAY[0..15,0..5] OF LREAL	MCS1 を基準にした UCS のデータ
UCS	ARRAY[0..15] OF _sMC_COORD_REF	UCS 設定データ
TCS	ARRAY[0..16] OF _sMC_TOOL_COORD_REF	TCS 設定データ
KinTransform	_sMC_KIN_REF	ロボットのキネマティクス設定データ
Workspace	_sMC_WORKSPACE_REF	ロボットのワークスペース設定データ
PickRange	ARRAY[0..16] OF sSimOpeRange	搬入コンベアの起動・見逃しラインに関するデータ群
PlaceRange	ARRAY[0..16] OF sSimOpeRange	搬出コンベアの起動・見逃しラインに関するデータ群

sSimOpeRange の仕様

構造体型 sSimOpeRange の仕様は、以下のとおりです

名称	データ型	内容
sSimOpeRange	STRUCT	
StartLineP1	ARRAY[0..2] OF LREAL	起動ライン 端点 1 の X,Y,Z 座標
StartLineP2	ARRAY[0..2] OF LREAL	起動ライン 端点 2 の X,Y,Z 座標
MissingLineP1	ARRAY[0..2] OF LREAL	見逃しライン 端点 1 の X,Y,Z 座標
MissingLineP2	ARRAY[0..2] OF LREAL	見逃しライン 端点 2 の X,Y,Z 座標

sSimLine の仕様

構造体型 sSimLine の仕様は、以下のとおりです

名称	データ型	内容
sSimLine	STRUCT	
AxisNumber	UINT	搬入コンベア、または、搬出コンベアの 3D 機構モデルに割り付けた軸の軸番号
UnitVector	ARRAY[0..2] OF LREAL	搬入コンベア、または、搬出コンベアの単位ベクトル
CameraSensor Origin	ARRAY[0..5] OF LREAL	搬入コンベアに設置されている画像センサ、または、搬出コンベアに設置されている ON/OFF センサの、MCS1 を基準とした原点座標
MeasureInterval	LREAL	搬入コンベアの場合、画像センサの撮影タイミング間隔 搬出コンベアの場合、箱の大きさ
MeasureInterval Offset	LREAL	搬入コンベアの場合、使用しない 搬出コンベアの場合、箱と箱の間隔
LineLength	LREAL	搬入コンベア、または、搬出コンベアのコンベア長さ
ConfigTable	ARRAY[0..7] OF sLineConfigTable	搬入コンベア、または、搬出コンベアのロボット構成情報に関するデータ群

sLineConfigTable の仕様

構造体型 sLineConfigTable の仕様は、以下のとおりです

名称	データ型	内容
sLineConfigTable	STRUCT	
RobotNumber	UINT	ロボット No.
CSID	UINT	ロボットの CSID
SensorType	UINT	搬入コンベア、または、搬出コンベアに設置されているセンサの種別 0: 未使用 1: 画像センサ 2: ON/OFF センサ
CameraSensor Number	UINT	搬入コンベアに設置されている画像センサの Line No. 搬出コンベアでは使用しません

sSimVision の仕様

構造体型 sSimVision の仕様は、以下のとおりです

名称	データ型	内容
sSimVision	STRUCT	
Origin	ARRAY[0..2] OF LREAL	MCS1 を基準にした画像センサの原点座標
Dx	LREAL	MCS1 を基準にしたコンベアの単位移動量(X 方向)
Dy	LREAL	MCS1 を基準にしたコンベアの単位移動量(Y 方向)
Length	LREAL	カメラ視野の長さ(X 視野)
Width	LREAL	カメラ視野の幅(Y 視野)
TiltAngle	LREAL	カメラ視野の傾き

3

パラメータの説明

この章では、キャリブレーションパラメータの個々のデータについて説明しています。

3-1	ロボットに関するデータ	3-2
3-2	搬入コンベアに関するデータ	3-10
3-3	搬出コンベアに関するデータ	3-14
3-4	画像センサに関するデータ	3-17
3-4-1	画像センサの通信コマンドマクロ例	3-19
3-4-2	CPU ユニットのプログラム例	3-25

3-1 ロボットに関するデータ

ロボットに関するキャリブレーションパラメータについて説明します。

ロボット1台につき、以下のキャリブレーションパラメータデータを代入するSTプログラムが出力されます。

キャリブレーションパラメータデータ	
SimData.Robot[1].UCSoverMCS1[α , 0]	1ライン目の搬入コンベアのデータです
SimData.Robot[1].UCSoverMCS1[α , 1]	
SimData.Robot[1].UCSoverMCS1[α , 2]	
SimData.Robot[1].UCSoverMCS1[α , 3]	
SimData.Robot[1].UCSoverMCS1[α , 4]	
SimData.Robot[1].UCSoverMCS1[α , 5]	
SimData.Robot[1].UCS[α].CSID	
SimData.Robot[1].UCS[α].Pose[0]	
SimData.Robot[1].UCS[α].Pose[1]	
SimData.Robot[1].UCS[α].Pose[2]	
SimData.Robot[1].UCS[α].Pose[3]	
SimData.Robot[1].UCS[α].Pose[4]	
SimData.Robot[1].UCS[α].Pose[5]	
SimData.Robot[1].UCSoverMCS1[β , 0]	2ライン目以降の搬入コンベア、 または、搬出コンベアのデータです 使用ライン数分出力されます
SimData.Robot[1].UCSoverMCS1[β , 1]	
SimData.Robot[1].UCSoverMCS1[β , 2]	
SimData.Robot[1].UCSoverMCS1[β , 3]	
SimData.Robot[1].UCSoverMCS1[β , 4]	
SimData.Robot[1].UCSoverMCS1[β , 5]	
SimData.Robot[1].UCS[β].CSID	
SimData.Robot[1].UCS[β].Pose[0]	
SimData.Robot[1].UCS[β].Pose[1]	
SimData.Robot[1].UCS[β].Pose[2]	
SimData.Robot[1].UCS[β].Pose[3]	
SimData.Robot[1].UCS[β].Pose[4]	
SimData.Robot[1].UCS[β].Pose[5]	
SimData.Robot[1].UCSoverMCS1[γ , 0]	
SimData.Robot[1].UCSoverMCS1[γ , 1]	
SimData.Robot[1].UCSoverMCS1[γ , 2]	
SimData.Robot[1].UCSoverMCS1[γ , 3]	
SimData.Robot[1].UCSoverMCS1[γ , 4]	
SimData.Robot[1].UCSoverMCS1[γ , 5]	
SimData.Robot[1].UCS[γ].CSID	
SimData.Robot[1].UCS[γ].Pose[0]	
SimData.Robot[1].UCS[γ].Pose[1]	
SimData.Robot[1].UCS[γ].Pose[2]	
SimData.Robot[1].UCS[γ].Pose[3]	
SimData.Robot[1].UCS[γ].Pose[4]	
SimData.Robot[1].UCS[γ].Pose[5]	

キャリブレーションパラメータデータ	
SimData.Robot[1].TCS[1].ToolID	1 個目のロボットツールのデータです
SimData.Robot[1].TCS[1].Pose[0]	
SimData.Robot[1].TCS[1].Pose[1]	
SimData.Robot[1].TCS[1].Pose[2]	
SimData.Robot[1].TCS[1].Pose[3]	
SimData.Robot[1].TCS[1].Pose[4]	
SimData.Robot[1].TCS[1].Pose[5]	
SimData.Robot[1].TCS[2].ToolID	2 個目以降のロボットツールのデータです
SimData.Robot[1].TCS[2].Pose[0]	
SimData.Robot[1].TCS[2].Pose[1]	
SimData.Robot[1].TCS[2].Pose[2]	
SimData.Robot[1].TCS[2].Pose[3]	
SimData.Robot[1].TCS[2].Pose[4]	
SimData.Robot[1].TCS[2].Pose[5]	
SimData.Robot[1].TCS[16].ToolID	2 個目以降のロボットツールのデータです
SimData.Robot[1].TCS[16].Pose[0]	
SimData.Robot[1].TCS[16].Pose[1]	
SimData.Robot[1].TCS[16].Pose[2]	
SimData.Robot[1].TCS[16].Pose[3]	
SimData.Robot[1].TCS[16].Pose[4]	
SimData.Robot[1].TCS[16].Pose[5]	
SimData.Robot[1].KinTransform.KinType	ロボットツールの使用個数分出力されます
SimData.Robot[1].KinTransform.KinParam[0]	
SimData.Robot[1].KinTransform.KinParam[1]	
SimData.Robot[1].KinTransform.KinParam[2]	
SimData.Robot[1].KinTransform.KinParam[3]	
SimData.Robot[1].KinTransform.ExpansionParam[0]	
SimData.Robot[1].Workspace.WorkspaceType	
SimData.Robot[1].Workspace.WorkspaceParam[0]	ロボットツールの使用個数分出力されます
SimData.Robot[1].Workspace.WorkspaceParam[1]	
SimData.Robot[1].Workspace.WorkspaceParam[2]	
SimData.Robot[1].Workspace.WorkspaceParam[3]	
SimData.Robot[1].Workspace.WorkspaceParam[4]	

キャリブレーションパラメータデータ	
SimData.Robot[1].PickRange[1].StartLineP1[0]	搬入コンベアの 1 ライン目のデータです
SimData.Robot[1].PickRange[1].StartLineP1[1]	
SimData.Robot[1].PickRange[1].StartLineP1[2]	
SimData.Robot[1].PickRange[1].StartLineP2[0]	
SimData.Robot[1].PickRange[1].StartLineP2[1]	
SimData.Robot[1].PickRange[1].StartLineP2[2]	
SimData.Robot[1].PickRange[1].MissingLineP1[0]	
SimData.Robot[1].PickRange[1].MissingLineP1[1]	
SimData.Robot[1].PickRange[1].MissingLineP1[2]	
SimData.Robot[1].PickRange[1].MissingLineP2[0]	
SimData.Robot[1].PickRange[1].MissingLineP2[1]	
SimData.Robot[1].PickRange[1].MissingLineP2[2]	
SimData.Robot[1].PickRange[2].StartLineP1[0]	搬入コンベアの 2 ライン目以降のデータです 搬入コンベアの使用ライン数分出力されます
SimData.Robot[1].PickRange[2].StartLineP1[1]	
SimData.Robot[1].PickRange[2].StartLineP1[2]	
SimData.Robot[1].PickRange[2].StartLineP2[0]	
SimData.Robot[1].PickRange[2].StartLineP2[1]	
SimData.Robot[1].PickRange[2].StartLineP2[2]	
SimData.Robot[1].PickRange[2].MissingLineP1[0]	
SimData.Robot[1].PickRange[2].MissingLineP1[1]	
SimData.Robot[1].PickRange[2].MissingLineP1[2]	
SimData.Robot[1].PickRange[2].MissingLineP2[0]	
SimData.Robot[1].PickRange[2].MissingLineP2[1]	
SimData.Robot[1].PickRange[2].MissingLineP2[2]	
SimData.Robot[1].PickRange[16].StartLineP1[0]	搬出コンベアの 1 ライン目のデータです
SimData.Robot[1].PickRange[16].StartLineP1[1]	
SimData.Robot[1].PickRange[16].StartLineP1[2]	
SimData.Robot[1].PickRange[16].StartLineP2[0]	
SimData.Robot[1].PickRange[16].StartLineP2[1]	
SimData.Robot[1].PickRange[16].StartLineP2[2]	
SimData.Robot[1].PickRange[16].MissingLineP1[0]	
SimData.Robot[1].PickRange[16].MissingLineP1[1]	
SimData.Robot[1].PickRange[16].MissingLineP1[2]	
SimData.Robot[1].PickRange[16].MissingLineP2[0]	
SimData.Robot[1].PickRange[16].MissingLineP2[1]	
SimData.Robot[1].PickRange[16].MissingLineP2[2]	
SimData.Robot[1].PlaceRange[1].StartLineP1[0]	搬出コンベアの 1 ライン目のデータです
SimData.Robot[1].PlaceRange[1].StartLineP1[1]	
SimData.Robot[1].PlaceRange[1].StartLineP1[2]	
SimData.Robot[1].PlaceRange[1].StartLineP2[0]	
SimData.Robot[1].PlaceRange[1].StartLineP2[1]	
SimData.Robot[1].PlaceRange[1].StartLineP2[2]	
SimData.Robot[1].PlaceRange[1].MissingLineP1[0]	
SimData.Robot[1].PlaceRange[1].MissingLineP1[1]	
SimData.Robot[1].PlaceRange[1].MissingLineP1[2]	
SimData.Robot[1].PlaceRange[1].MissingLineP2[0]	
SimData.Robot[1].PlaceRange[1].MissingLineP2[1]	
SimData.Robot[1].PlaceRange[1].MissingLineP2[2]	

キャリブレーションパラメータデータ	
SimData.Robot[1].PlaceRange[2].StartLineP1[0]	
SimData.Robot[1].PlaceRange[2].StartLineP1[1]	
SimData.Robot[1].PlaceRange[2].StartLineP1[2]	
SimData.Robot[1].PlaceRange[2].StartLineP2[0]	
SimData.Robot[1].PlaceRange[2].StartLineP2[1]	
SimData.Robot[1].PlaceRange[2].StartLineP2[2]	
SimData.Robot[1].PlaceRange[2].MissingLineP1[0]	
SimData.Robot[1].PlaceRange[2].MissingLineP1[1]	
SimData.Robot[1].PlaceRange[2].MissingLineP1[2]	
SimData.Robot[1].PlaceRange[2].MissingLineP2[0]	
SimData.Robot[1].PlaceRange[2].MissingLineP2[1]	
SimData.Robot[1].PlaceRange[2].MissingLineP2[2]	
SimData.Robot[1].PlaceRange[16].StartLineP1[0]	
SimData.Robot[1].PlaceRange[16].StartLineP1[1]	
SimData.Robot[1].PlaceRange[16].StartLineP1[2]	
SimData.Robot[1].PlaceRange[16].StartLineP2[0]	
SimData.Robot[1].PlaceRange[16].StartLineP2[1]	
SimData.Robot[1].PlaceRange[16].StartLineP2[2]	
SimData.Robot[1].PlaceRange[16].MissingLineP1[0]	
SimData.Robot[1].PlaceRange[16].MissingLineP1[1]	
SimData.Robot[1].PlaceRange[16].MissingLineP1[2]	
SimData.Robot[1].PlaceRange[16].MissingLineP2[0]	
SimData.Robot[1].PlaceRange[16].MissingLineP2[1]	
SimData.Robot[1].PlaceRange[16].MissingLineP2[2]	

搬出コンベアの 2 ライン目以降のデータです
搬出コンベアの使用ライン数分出力されます

※ α 、 β 、 γ には 0~15 の中の任意の数値が付けられます。

2 台目以降のロボットのデータは、以下のとおりです。

SimData.Robot[2].*****	: 2 台目のロボットのデータ
SimData.Robot[3].*****	: 3 台目のロボットのデータ
SimData.Robot[4].*****	: 4 台目のロボットのデータ

SimData.Robot[1].UCSoverMCS1[α ,0-5]の説明

MCS1 を基準にした UCS の値が代入されます。

UCS で表された位置を MCS1 の位置に変換する場合、あるいは、MCS1 で表された位置を UCS に変換する場合に、本変数を使用して変換してください。

変換のプログラム例は省略します。

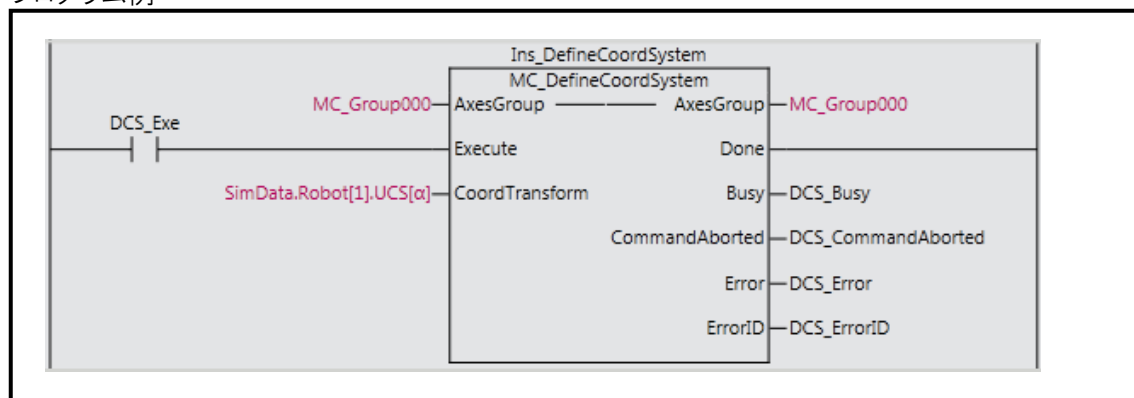
SimData.Robot[1].UCS[α].CSID

SimData.Robot[1].UCS[α].Pose[0-5]の説明

ユーザ座標系(UCS)を定義する値が代入されます。

MC_DefineCoordSystem(座標系定義)命令を使用して、ユーザ座標系(UCS)を設定してください。

プログラム例



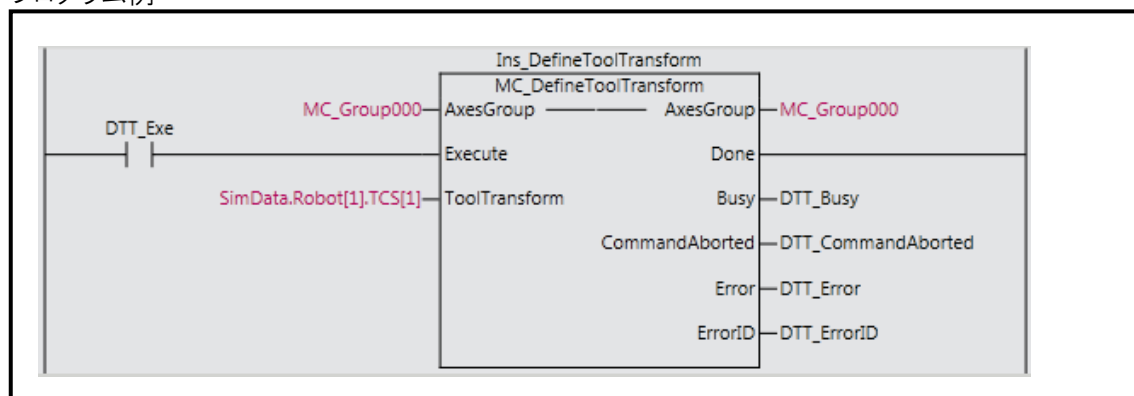
SimData.Robot[1].TCS[1].ToolID

SimData.Robot[1].TCS[1].Pose[0-5]の説明

ツール座標系(TCS)を定義する値が代入されます。

MC_DefineToolTransform(ツール座標系定義)命令を使用して、ツール座標系(TCS)を設定してください。

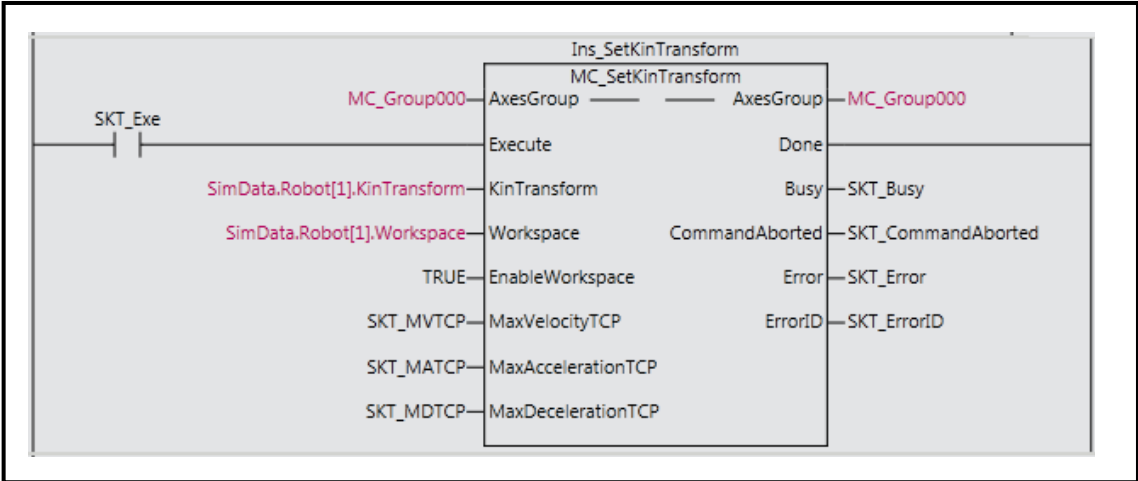
プログラム例



- SimData.Robot[1].KinTransform.KinType
- SimData.Robot[1].KinTransform.KinParam[0-3]
- SimData.Robot[1].KinTransform.ExpansionParam[0]の説明

キネマティクスを設定するための値(パラメータ)が代入されます。
MC_SetKinTransform (キネマティクス変換設定)命令を使用して、キネマティクス変換パラメータを設定してください。

プログラム例



- SimData.Robot[1].Workspace.WorkspaceType
- SimData.Robot[1].Workspace.WorkspaceParam[0-4]の説明

ワークスペースチェックを行うためのワークスペースを設定する値(パラメータ)が代入されます。
MC_SetKinTransform (キネマティクス変換設定)命令を使用して、ワークスペースを設定してください。

プログラム例については、「SimData.Robot[1].KinTransform.の説明」を参照してください。

SimData.Robot[1].PickRange[1].StartLineP1[0-2]**SimData.Robot[1].PickRange[1].StartLineP2[0-2]の説明**

起動ラインを定義するための値(二つの端点の座標値)が代入されます。

座標値は各ロボットの機械座標系(MCS)です。

- ・ SimData.Robot[1].PickRange[1].StartLineP1[0] : 一つ目の端点の X 座標
- ・ SimData.Robot[1].PickRange[1].StartLineP1[1] : 一つ目の端点の Y 座標
- ・ SimData.Robot[1].PickRange[1].StartLineP1[2] : 一つ目の端点の Z 座標
- ・ SimData.Robot[1].PickRange[1].StartLineP2[0] : 二つ目の端点の X 座標
- ・ SimData.Robot[1].PickRange[1].StartLineP2[1] : 二つ目の端点の Y 座標
- ・ SimData.Robot[1].PickRange[1].StartLineP2[2] : 二つ目の端点の Z 座標

搬入ワークの現在位置が、二つの端点で定義された直線(起動ライン)を超えた場合に、ロボットの Pick 動作を開始するプログラムを作成してください。

搬入ワークの現在位置が、ユーザ座標系(UCS)やロボット 1 の機械座標系(MCS1)の場合は、本変数の値を、搬入ワークの現在位置の座標系に変換して使用してください。

プログラム例は省略します。

SimData.Robot[1].PickRange[1].MissingLineP1[0-2]**SimData.Robot[1].PickRange[1].MissingLineP2[0-2]の説明**

搬入コンベアの見逃しラインを定義するための値(二つの端点の座標値)が代入されます。

座標値は各ロボットの機械座標系(MCS)です。

- ・ SimData.Robot[1].PickRange[1].MissingLineP1[0] : 一つ目の端点の X 座標
- ・ SimData.Robot[1].PickRange[1].MissingLineP1[1] : 一つ目の端点の Y 座標
- ・ SimData.Robot[1].PickRange[1].MissingLineP1[2] : 一つ目の端点の Z 座標
- ・ SimData.Robot[1].PickRange[1].MissingLineP2[0] : 二つ目の端点の X 座標
- ・ SimData.Robot[1].PickRange[1].MissingLineP2[1] : 二つ目の端点の Y 座標
- ・ SimData.Robot[1].PickRange[1].MissingLineP2[2] : 二つ目の端点の Z 座標

搬入ワークの現在位置が、二つの端点で定義された直線(見逃しライン)を超えた場合に、ロボットの Pick 動作を開始しないプログラムを作成してください。

搬入ワークの現在位置が、ユーザ座標系(UCS)やロボット 1 の機械座標系(MCS1)の場合は、本変数の値を、搬入ワークの現在位置の座標系に変換して使用してください。

プログラム例は省略します。

SimData.Robot[1].PlaceRange[1].StartLineP1[0-2]**SimData.Robot[1].PlaceRange[1].StartLineP2[0-2]の説明**

起動ラインを定義するための値(二つの端点の座標値)が代入されます。

座標値は各ロボットの機械座標系(MCS)です。

- ・ SimData.Robot[1].PlaceRange[1].StartLineP1[0] : 一つ目の端点の X 座標
- ・ SimData.Robot[1].PlaceRange[1].StartLineP1[1] : 一つ目の端点の Y 座標
- ・ SimData.Robot[1].PlaceRange[1].StartLineP1[2] : 一つ目の端点の Z 座標
- ・ SimData.Robot[1].PlaceRange[1].StartLineP2[0] : 二つ目の端点の X 座標
- ・ SimData.Robot[1].PlaceRange[1].StartLineP2[1] : 二つ目の端点の Y 座標
- ・ SimData.Robot[1].PlaceRange[1].StartLineP2[2] : 二つ目の端点の Z 座標

搬出ワークを置くための、搬出コンベアの区切線、あるいは、搬出コンベアの箱の現在位置が、二つの端点で定義された直線(起動ライン)を超えた場合に、ロボットの Place 動作を開始するプログラムを作成してください。

搬出ワークを置くための位置が、ユーザ座標系(UCS)やロボット 1 の機械座標系(MCS1)の場合は、本変数の値を、搬出ワークを置くための位置の座標系に変換して使用してください。

プログラム例は省略します。

SimData.Robot[1].PlaceRange[1].MissingLineP1[0-2]**SimData.Robot[1].PlaceRange[1].MissingLineP2[0-2]の説明**

搬入コンベアの見逃しラインを定義するための値(二つの端点の座標値)が代入されます。

座標値は各ロボットの機械座標系(MCS)です。

- ・ SimData.Robot[1].PlaceRange[1].MissingLineP1[0] : 一つ目の端点の X 座標
- ・ SimData.Robot[1].PlaceRange[1].MissingLineP1[1] : 一つ目の端点の Y 座標
- ・ SimData.Robot[1].PlaceRange[1].MissingLineP1[2] : 一つ目の端点の Z 座標
- ・ SimData.Robot[1].PlaceRange[1].MissingLineP2[0] : 二つ目の端点の X 座標
- ・ SimData.Robot[1].PlaceRange[1].MissingLineP2[1] : 二つ目の端点の Y 座標
- ・ SimData.Robot[1].PlaceRange[1].MissingLineP2[2] : 二つ目の端点の Z 座標

搬出ワークを置くための、搬出コンベアの区切線、あるいは、搬出コンベアの箱の現在位置が、二つの端点で定義された直線(見逃しライン)を超えた場合に、ロボットの Place 動作を開始しないプログラムを作成してください。

搬出ワークを置くための位置が、ユーザ座標系(UCS)やロボット 1 の機械座標系(MCS1)の場合は、本変数の値を、搬出ワークを置くための位置の座標系に変換して使用してください。

プログラム例は省略します。

3-2 搬入コンベアに関するデータ

搬入コンベアに関するキャリブレーションパラメータについて説明します。

搬入コンベア 1 ラインにつき、以下のキャリブレーションパラメータデータを代入する ST プログラムが出力されます。

キャリブレーションパラメータデータ	
SimData.PickLine[1].AxisNumber	
SimData.PickLine[1].UnitVector[0]	
SimData.PickLine[1].UnitVector[1]	
SimData.PickLine[1].UnitVector[2]	
SimData.PickLine[1].CameraSensorOrigin[0]	
SimData.PickLine[1].CameraSensorOrigin[1]	
SimData.PickLine[1].CameraSensorOrigin[2]	
SimData.PickLine[1].CameraSensorOrigin[3]	
SimData.PickLine[1].CameraSensorOrigin[4]	
SimData.PickLine[1].CameraSensorOrigin[5]	
SimData.PickLine[1].MeasureInterval	
SimData.PickLine[1].MeasureIntervalOffset	
SimData.PickLine[1].LineLength	
SimData.PickLine[1].ConfigTable[0].RobotNumber	1 台目のロボットのデータです
SimData.PickLine[1].ConfigTable[0].CSID	
SimData.PickLine[1].ConfigTable[0].SensorType	
SimData.PickLine[1].ConfigTable[0].CameraSensorNumber	
SimData.PickLine[1].ConfigTable[1].RobotNumber	2 台目以降のロボットのデータです
SimData.PickLine[1].ConfigTable[1].CSID	
SimData.PickLine[1].ConfigTable[2].RobotNumber	
SimData.PickLine[1].ConfigTable[2].CSID	
	ロボットの使用台数分出力されます
SimData.PickLine[1].ConfigTable[7].RobotNumber	
SimData.PickLine[1].ConfigTable[7].CSID	

2 ライン目以降の搬入コンベアのデータは、以下のとおりです。

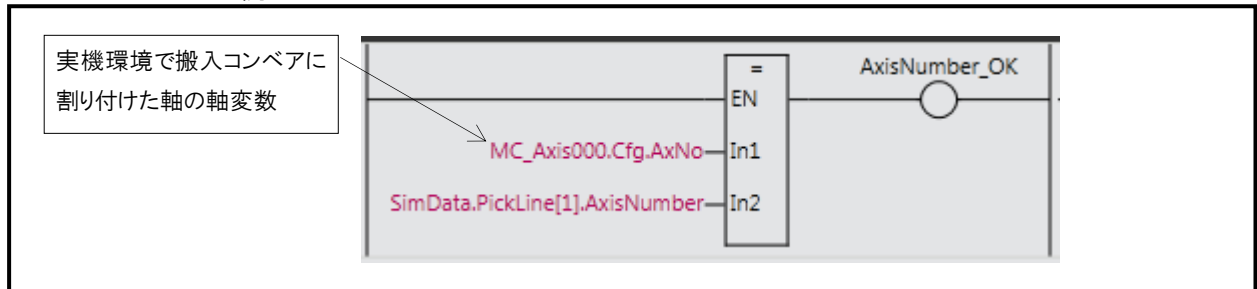
SimData.PickLine[2].*****	: 2 ライン目の搬入コンベアのデータ
SimData.PickLine[3].*****	: 3 ライン目の搬入コンベアのデータ
SimData.PickLine[4].*****	: 4 ライン目の搬入コンベアのデータ

SimData.PickLine[1].AxisNumber の説明

3D 機構モデルの搬入コンベアに割り付けた軸の軸番号が代入されます。

実機環境で搬入コンベアに割り付けた軸の軸変数から取得できる軸番号と、本変数とが一致していることを確認する、インターロックプログラムを作成してください。

プログラム例



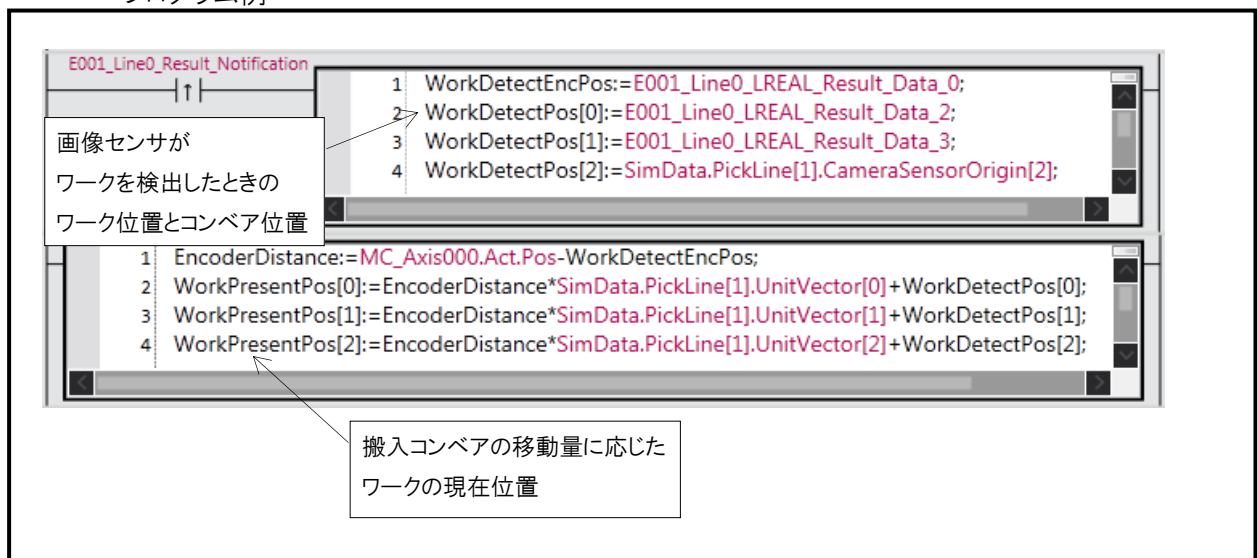
SimData.PickLine[1].UnitVector[0-2]の説明

搬入コンベアの単位ベクトル(MCS1 を基準にした搬入コンベアの進行方向)が代入されます。

- ・ SimData.PickLine[1].UnitVector[0] : 単位ベクトルの X 軸方向成分
- ・ SimData.PickLine[1].UnitVector[1] : 単位ベクトルの Y 軸方向成分
- ・ SimData.PickLine[1].UnitVector[2] : 単位ベクトルの Z 軸方向成分

画像センサが検出したワーク位置、および、搬入コンベアの移動量と、本変数から、ワークの現在位置を算出するプログラムを作成してください。

プログラム例



SimData.PickLine[1].CameraSensorOrigin[0-5]の説明

搬入コンベアに設置されている画像センサの、MCS1 を基準とした原点座標が代入されます。

- ・ SimData.PickLine[1].CameraSensorOrigin[0] : X 座標
- ・ SimData.PickLine[1].CameraSensorOrigin[1] : Y 座標
- ・ SimData.PickLine[1].CameraSensorOrigin[2] : Z 座標
- ・ SimData.PickLine[1].CameraSensorOrigin[3] : Rx 座標
- ・ SimData.PickLine[1].CameraSensorOrigin[4] : Ry 座標
- ・ SimData.PickLine[1].CameraSensorOrigin[5] : Rz 座標

画像センサが検出したワーク位置の Z 座標値を決める際に、本変数の Z 座標の値を使用するプログラムを作成してください。

プログラム例については、「SimData.PickLine[1].UnitVector[0-2]の説明」を参照してください。

SimData.PickLine[1].MeasureInterval の説明

画像センサの撮影タイミング間隔が代入されます。

搬入コンベアの現在位置が、本変数の値の移動を繰り返す毎に、画像センサの計測トリガを ON するプログラムを作成してください。

プログラム例

```
1 IF MC_Axis000.Act.Pos >= TriggerEncPos + SimData.PickLine[1].MeasureInterval THEN
2   E001_Line0_Trigger := TRUE;
3   TriggerEncPos := MC_Axis000.Act.Pos;
4 END_IF;
```

SimData.PickLine[1].MeasureIntervalOffset の説明

コンベアが搬出コンベアのとくに使用するキャリブレーションパラメータです。

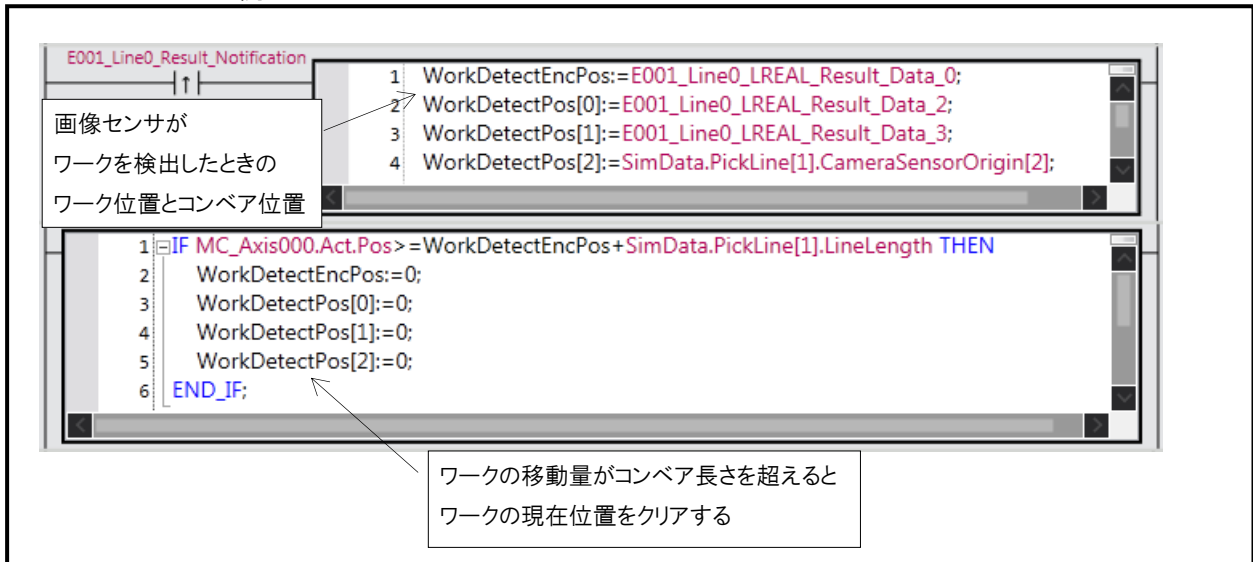
搬入コンベアでは、このキャリブレーションパラメータは使用しません。

SimData.PickLine[1].LineLength の説明

搬入コンベアのコンベア長さが代入されます。

画像センサが検出したワークの移動量が、本変数の値を超えた場合に、該当ワークの現在位置算出を終わらせるプログラムを作成してください。

プログラム例



SimData.PickLine[1].ConfigTable[0].RobotNumber の説明

搬入コンベアの 1 台目のロボットのロボット No.が代入されます。

本変数に関するプログラムは不要です。

SimData.PickLine[1].ConfigTable[0].CSID の説明

搬入コンベアの 1 台目のロボットの CSID が代入されます。

本変数に関するプログラムは不要です。

SimData.PickLine[1].ConfigTable[0].SensorType の説明

搬入コンベアに設置されているセンサの種別が代入されます。

搬入コンベアのセンサ種別は 1:画像センサ のみに対応しています。常に 1 が代入されます。

本変数に関するプログラムは不要です。

SimData.PickLine[1].ConfigTable[0].CameraSensorNumber の説明

搬入コンベアに設置されているセンサ No.(画像センサの Line No.)が代入されます。

本変数に関するプログラムは不要です。

3-3 搬出コンベアに関するデータ

搬出コンベアに関するキャリブレーションパラメータについて説明します。

搬出コンベア 1 ラインにつき、以下のキャリブレーションパラメータデータを代入する ST プログラムが出力されます。

キャリブレーションパラメータデータ	
SimData.PlaceLine[1].AxisNumber	
SimData.PlaceLine[1].UnitVector[0]	
SimData.PlaceLine[1].UnitVector[1]	
SimData.PlaceLine[1].UnitVector[2]	
SimData.PlaceLine[1].CameraSensorOrigin[0]	
SimData.PlaceLine[1].CameraSensorOrigin[1]	
SimData.PlaceLine[1].CameraSensorOrigin[2]	
SimData.PlaceLine[1].CameraSensorOrigin[3]	
SimData.PlaceLine[1].CameraSensorOrigin[4]	
SimData.PlaceLine[1].CameraSensorOrigin[5]	
SimData.PlaceLine[1].MeasureInterval	
SimData.PlaceLine[1].MeasureIntervalOffset	
SimData.PlaceLine[1].LineLength	
SimData.PlaceLine[1].ConfigTable[0].RobotNumber	1 台目のロボットのデータです
SimData.PlaceLine[1].ConfigTable[0].CSID	
SimData.PlaceLine[1].ConfigTable[0].SensorType	
SimData.PlaceLine[1].ConfigTable[0].CameraSensorNumber	
SimData.PlaceLine[1].ConfigTable[1].RobotNumber	2 台目以降のロボットのデータです ロボットの使用台数分出力されます
SimData.PlaceLine[1].ConfigTable[1].CSID	
SimData.PlaceLine[1].ConfigTable[7].RobotNumber	
SimData.PlaceLine[1].ConfigTable[7].CSID	

2 ライン目以降の搬出コンベアのデータは、以下のとおりです。

- SimData.PlaceLine[2].***** :2 ライン目の搬出コンベアのデータ
- SimData.PlaceLine[3].***** :3 ライン目の搬出コンベアのデータ
- SimData.PlaceLine[4].***** :4 ライン目の搬出コンベアのデータ
-
-
-

搬出コンベアに関するデータのうち、以下のデータは、搬入コンベアと同じ内容のデータが出力されます。データの説明やプログラム例は、「3-2 搬入コンベアに関するデータ」を参照してください。

- ・ SimData.PlaceLine[1].AxisNumber
- ・ SimData.PlaceLine[1].UnitVector[0]
- ・ SimData.PlaceLine[1].UnitVector[1]
- ・ SimData.PlaceLine[1].UnitVector[2]
- ・ SimData.PlaceLine[1].CameraSensorOrigin[0]
- ・ SimData.PlaceLine[1].CameraSensorOrigin[1]
- ・ SimData.PlaceLine[1].CameraSensorOrigin[2]
- ・ SimData.PlaceLine[1].CameraSensorOrigin[3]
- ・ SimData.PlaceLine[1].CameraSensorOrigin[4]
- ・ SimData.PlaceLine[1].CameraSensorOrigin[5]
- ・ SimData.PlaceLine[1].LineLength
- ・ SimData.PlaceLine[1].ConfigTable[0].RobotNumber
- ・ SimData.PlaceLine[1].ConfigTable[0].CSID
- ・ SimData.PlaceLine[1].ConfigTable[1].RobotNumber
- ・ SimData.PlaceLine[1].ConfigTable[1].CSID
- ・ |
- ・ |
- ・ |
- ・ SimData.PlaceLine[1].ConfigTable[7].RobotNumber
- ・ SimData.PlaceLine[1].ConfigTable[7].CSID

以下のデータは、搬入コンベアと異なる内容のため、データの説明やプログラム例を後述します。

- ・ SimData.PlaceLine[1].MeasureInterval
- ・ SimData.PlaceLine[1].MeasureIntervalOffset
- ・ SimData.PlaceLine[1].ConfigTable[0].SensorType

以下のデータは、搬出コンベアの場合に出力されません。

- ・ SimData.PlaceLine[1].ConfigTable[0].CameraSensorNumber

SimData.PlaceLine[1].MeasureInterval の説明

搬出コンベアの種別が「搬出用コンベア(区切りあり)」の場合は「境界線:間隔」の値が、搬出コンベアの種別が「搬出用コンベア(箱)」の場合は「箱:長さ」の値が代入されます。

搬出コンベアの現在位置が、本変数と SimData.PlaceLine[1].MeasureIntervalOffset 変数を合わせた値の移動を繰り返す毎に、センサ入力トリガを ON するプログラムを作成してください。

プログラム例

```
1 IF MC_Axis001.Act.Pos >= TriggerEncPos + SimData.PlaceLine[1].MeasureInterval
2   + SimData.PlaceLine[1].MeasureIntervalOffset THEN
3   PlaceLine_Trigger:=TRUE;
4   TriggerEncPos:=MC_Axis001.Act.Pos;
5 END_IF;
```

SimData.PlaceLine[1].MeasureIntervalOffset の説明

搬出コンベアの種別が「搬出用コンベア(区切りあり)」の場合は固定値が、搬出コンベアの種別が「搬出用コンベア(箱)」の場合は「箱:間隔」の値が代入されます。

搬出コンベアの現在位置が、本変数と SimData.PlaceLine[1].MeasureInterval 変数を合わせた値の移動を繰り返す毎に、センサ入力トリガを ON するプログラムを作成してください。

プログラム例については、「SimData.PlaceLine[1].MeasureInterval の説明」を参照してください。

SimData.PlaceLine[1].ConfigTable[0].SensorType の説明

搬出コンベアに設置されているセンサの種別が代入されます。

搬出コンベアのセンサ種別は 2:ON/OFF センサ のみに対応しています。常に 2 が代入されます。

本変数に関するプログラムは不要です。

3-4 画像センサに関するデータ

画像センサに関するキャリブレーションパラメータについて説明します。

画像センサ 1 台につき、以下のキャリブレーションパラメータデータを代入する ST プログラムが出力されます。

キャリブレーションパラメータデータ	
SimData.Vision[0].Origin[0]	
SimData.Vision[0].Origin[1]	
SimData.Vision[0].Origin[2]	
SimData.Vision[0].Dx	
SimData.Vision[0].Dy	
SimData.Vision[0].Length	
SimData.Vision[0].Width	
SimData.Vision[0].TiltAngle	

2 台目以降の画像センサのデータは、以下のとおりです。

SimData.Vision[1].*****	:2 台目の画像センサのデータ
SimData.Vision[2].*****	:3 台目の画像センサのデータ
SimData.Vision[3].*****	:4 台目の画像センサのデータ

画像センサに関するキャリブレーションパラメータデータは、画像センサの通信コマンドを介して、処理ユニットデータに値を設定してください。

画像センサの通信コマンドマクロ例、および、CPU ユニットのプログラム例は、後述します。

SimData.Vision[0].Origin[0-2]の説明

MCS1 を基準にした画像センサの原点座標が代入されます。

- ・ SimData.Vision[0].Origin[0] : X 座標
- ・ SimData.Vision[0].Origin[1] : Y 座標
- ・ SimData.Vision[0].Origin[2] : Z 座標

SimData.Vision[0].Dx

SimData.Vision[0].Dy の説明

MCS1 を基準にしたコンベアの単位移動量が代入されます。

- ・ SimData.Vision[0].Dx : 単位移動量の X 軸方向成分
- ・ SimData.Vision[0].Dy : 単位移動量の Y 軸方向成分

SimData.Vision[0].Length

SimData.Vision[0].Width の説明

画像センサの視野の大きさ(長さ&幅)が代入されます。

- ・ SimData.Vision[0].Length : 画像センサの視野の長さ(X 方向の視野)
- ・ SimData.Vision[0].Width : 画像センサの視野の幅(Y 方向の視野)

SimData.Vision[0].TiltAngle の説明

画像センサの視野の傾きが代入されます。

3-4-1 画像センサの通信コマンドマクロ例

CPU ユニットから送信されるキャリブレーションパラメータを、画像センサの処理ユニットデータに設定する、通信コマンドマクロ例を説明します。

コマンド名	CMD0128
関数名	FUNC_0128
BUSY ON	ON

```
Dim READ_DATA&(7)
Dim BUF&(1)

COMMUNICATION_AREA& = 0
CHANNEL_OFFSET& = 10
CHANNEL_COUNT& = 16

PI# = 3.14159265358979

ERROR& = 0

GetSystemData "IoModule", "ioIdent3", IO_MODULE$
If UCase$(IO_MODULE$) = "ETHERCAT" Then

  GetSystemData IO_MODULE$, "_userInputDataSize", IO_SIZE&

  If IO_SIZE& = -1 Then
    GetSystemData IO_MODULE$, "userInputDataSize0", IO_SIZE&
  EndIf

  If IO_SIZE& <> 1048592 Then
    RESPONSECODE& = -1
    RESPONSEVALUE&(0) = 0
    Return
  EndIf

Else
  RESPONSECODE& = -1
  RESPONSEVALUE&(0) = 0
  Return
EndIf

If ARGUMENTVALUE#(0) = 0 Then

  I& = 0

  Do
    Try
      GetImageWindow I&, TMP_X&, TMP_Y&, TMP_WIDTH&, TMP_HEIGHT&, TMP_UNIT_NO&,
      TMP_SUB_NO&, TMP_MAG&, TMP_ORIGN_X&, TMP_ORIGN_Y&, TMP_UPDATE&, TMP_VISIBLE&
      SetImageWindow I&, TMP_X&, TMP_Y&, TMP_WIDTH&, TMP_HEIGHT&, TMP_UNIT_NO&,
      TMP_SUB_NO&, TMP_MAG&, TMP_ORIGN_X&, TMP_ORIGN_Y&, 0, TMP_VISIBLE&
    Catch
      Exit Do
    End Try

    I& = I& + 1
  Loop While (1)
```

```

RESPONSECODE& = 0
RESPONSEVALUE&(0) = 0

Elseif ARGUMENTVALUE#(0) = 1 Then

    FOV_ORIGN_X# = 0
    FOV_ORIGN_Y# = 0

    CONVEYOR_MOVE_X# = 0
    CONVEYOR_MOVE_Y# = 0

    FOV_WIDTH# = 0
    FOV_HEIGHT# = 0

    FOV_ANGLE# = 0

    Dim ERROR_FLAG&(4)
    For I& = 0 To 4
        ERROR_FLAG&(I&) = 1
    Next

    RESPONSECODE& = 0
    RESPONSEVALUE&(0) = 0

Elseif ARGUMENTVALUE#(0) = 2 Then

    ReadPlcMemory IO_MODULE$, COMMUNICATION_AREA&, CHANNEL_OFFSET&,
CHANNEL_COUNT&, READ_DATA&()

    BUF&(0) = READ_DATA&(4)
    BUF&(1) = READ_DATA&(5)
    GetPlcData IO_MODULE$, BUF&(), 0, 8, BUF#
    FOV_ORIGN_X# = BUF#

    BUF&(0) = READ_DATA&(6)
    BUF&(1) = READ_DATA&(7)
    GetPlcData IO_MODULE$, BUF&(), 0, 8, BUF#
    FOV_ORIGN_Y# = BUF#

    ERROR_FLAG&(0) = 0

    RESPONSECODE& = 0
    RESPONSEVALUE&(0) = 0

Elseif ARGUMENTVALUE#(0) = 3 Then

    ReadPlcMemory IO_MODULE$, COMMUNICATION_AREA&, CHANNEL_OFFSET&,
CHANNEL_COUNT&, READ_DATA&()

    BUF&(0) = READ_DATA&(4)
    BUF&(1) = READ_DATA&(5)
    GetPlcData IO_MODULE$, BUF&(), 0, 8, BUF#
    CONVEYOR_MOVE_X# = BUF#

    BUF&(0) = READ_DATA&(6)
    BUF&(1) = READ_DATA&(7)
    GetPlcData IO_MODULE$, BUF&(), 0, 8, BUF#
    CONVEYOR_MOVE_Y# = BUF#

```

```

ERROR_FLAG&(1) = 0

RESPONSECODE& = 0
RESPONSEVALUE&(0) = 0

Elseif ARGUMENTVALUE#(0) = 4 Then

    ReadPlcMemory IO_MODULE$, COMMUNICATION_AREA&, CHANNEL_OFFSET&,
CHANNEL_COUNT&, READ_DATA&()

    BUF&(0) = READ_DATA&(4)
    BUF&(1) = READ_DATA&(5)
    GetPlcData IO_MODULE$, BUF&(), 0, 8, BUF#
    FOV_WIDTH# = BUF#

    BUF&(0) = READ_DATA&(6)
    BUF&(1) = READ_DATA&(7)
    GetPlcData IO_MODULE$, BUF&(), 0, 8, BUF#
    FOV_HEIGHT# = BUF#

    ERROR_FLAG&(2) = 0

    RESPONSECODE& = 0
    RESPONSEVALUE&(0) = 0

Elseif ARGUMENTVALUE#(0) = 5 Then

    ReadPlcMemory IO_MODULE$, COMMUNICATION_AREA&, CHANNEL_OFFSET&,
CHANNEL_COUNT&, READ_DATA&()

    BUF&(0) = READ_DATA&(4)
    BUF&(1) = READ_DATA&(5)
    GetPlcData IO_MODULE$, BUF&(), 0, 8, BUF#
    FOV_ANGLE# = BUF#

    ERROR_FLAG&(3) = 0

    RESPONSECODE& = 0
    RESPONSEVALUE&(0) = 0

Elseif ARGUMENTVALUE#(0) = 6 Then

    If (ERROR_FLAG&(2) = 0) Then
        If (FOV_WIDTH# <= 0) OR (FOV_HEIGHT# <= 0) Then
            For I& = 0 To 4
                ERROR& = ERROR& + ERROR_FLAG&(I&) * 2 ^ I&
            Next
            RESPONSECODE& = -1
            RESPONSEVALUE&(0) = ERROR&
            Return
        EndIf
    EndIf

    If (ERROR_FLAG&(3) = 0) Then

        If 0 <= FOV_ANGLE# AND FOV_ANGLE# <= 3 Then
            If CONVEYOR_MOVE_X# <> 0 Then

```

```

    ANGLE# = Atn(CONVEYOR_MOVE_Y# / CONVEYOR_MOVE_X#) * 180 / PI# + FOV_ANGLE# * 90
    If CONVEYOR_MOVE_X# < 0 Then
        ANGLE# = ANGLE# + 180
    EndIf
    If CONVEYOR_MOVE_X# > 0 AND CONVEYOR_MOVE_Y# < 0 Then
        ANGLE# = ANGLE# + 360
    EndIf
Else
    ANGLE# = CONVEYOR_MOVE_Y# * 90 + FOV_ANGLE# * 90
    If CONVEYOR_MOVE_Y# < 0 Then
        ANGLE# = ANGLE# + 360
    EndIf
EndIf
Else
    For I& = 0 To 4
        ERROR& = ERROR& + ERROR_FLAG&(I&) * 2 ^ I&
    Next

    RESPONSECODE& = -1
    RESPONSEVALUE&(0) = ERROR&
    Return
EndIf

If ANGLE# > 180 Then
    RADIANT# = (ANGLE# - 360) / 180 * PI#
Elseif ANGLE# <= -180 Then
    ANGLE# =
    RADIANT# = (ANGLE# + 360) / 180 * PI#
Else
    RADIANT# = ANGLE# / 180 * PI#
EndIf

RBT_X0# = FOV_ORIGN_X#
RBT_X1# = FOV_ORIGN_X# + Cos(RADIANT#) * FOV_WIDTH#
RBT_X2# = FOV_ORIGN_X# - Sin(RADIANT#) * (-FOV_HEIGHT#)
RBT_X3# = FOV_ORIGN_X# + Cos(RADIANT#) * FOV_WIDTH# - Sin(RADIANT#) * (-FOV_HEIGHT#)
RBT_Y0# = FOV_ORIGN_Y#
RBT_Y1# = FOV_ORIGN_Y# + Sin(RADIANT#) * FOV_WIDTH#
RBT_Y2# = FOV_ORIGN_Y# + Cos(RADIANT#) * (-FOV_HEIGHT#)
RBT_Y3# = FOV_ORIGN_Y# + Sin(RADIANT#) * FOV_WIDTH# + Cos(RADIANT#) * (-FOV_HEIGHT#)

EndIf

I& = 0
COUNTER& = 0
CAM_SIZEX& = 0
CAM_SIZEY& = 0
RET& = 0

Do
    If UnitItemIdent$(I&) = "ConveyorCalibration" Then

        COUNTER& = COUNTER& + 1
        Try
            GetImageSize I&, 0, CAM_SIZEX&, CAM_SIZEY&
        Catch
            RET& = -1
        End Try
    End Try

```

```

If (CAM_SIZEX <= 0) OR (CAM_SIZEY <= 0) OR (RET = -1) Then

    For I = 0 To 4
        ERROR = ERROR + ERROR_FLAG(I) * 2 ^ I
    Next

    RESPONSECODE = -1
    RESPONSEVALUE(0) = ERROR
    Return
Else
    Exit Do
EndIf

Elseif UnitItemIdent$(I) = "" Then
    Exit Do
EndIf

I = I + 1
Loop While (1)

If COUNTER = 0 Then
    For I = 0 To 4
        ERROR = ERROR + ERROR_FLAG(I) * 2 ^ I
    Next

    RESPONSECODE = -1
    RESPONSEVALUE(0) = ERROR
    Return
EndIf

ERROR_FLAG(4) = 0

If (ERROR_FLAG(0) = 1) OR (ERROR_FLAG(1) = 1) OR (ERROR_FLAG(2) = 1) OR
(ERROR_FLAG(3) = 1) Then
    For I = 0 To 4
        ERROR = ERROR + ERROR_FLAG(I) * 2 ^ I
    Next

    RESPONSECODE = -1
    RESPONSEVALUE(0) = ERROR
    Return
EndIf

I = 0

Do
    If UnitItemIdent$(I) = "ConveyorCalibration" Then

        SetUnitData I, "samplingPointNum", 4
        SetUnitData I, "calcMethod", 1

        SetUnitData I, "cameraX0", 0
        SetUnitData I, "cameraX1", CAM_SIZEX
        SetUnitData I, "cameraX2", 0
        SetUnitData I, "cameraX3", CAM_SIZEX
        SetUnitData I, "cameraY0", 0
        SetUnitData I, "cameraY1", 0
        SetUnitData I, "cameraY2", CAM_SIZEY
        SetUnitData I, "cameraY3", CAM_SIZEY
    
```

```
SetUnitData I&, "robotX0", RBT_X0#
SetUnitData I&, "robotX1", RBT_X1#
SetUnitData I&, "robotX2", RBT_X2#
SetUnitData I&, "robotX3", RBT_X3#
SetUnitData I&, "robotY0", RBT_Y0#
SetUnitData I&, "robotY1", RBT_Y1#
SetUnitData I&, "robotY2", RBT_Y2#
SetUnitData I&, "robotY3", RBT_Y3#

SetUnitData I&, "cameraEncCount", 0
SetUnitData I&, "robotEncCount1", 0
SetUnitData I&, "robotEncCount2", 0

SetUnitData I&, "encDistX", CONVEYOR_MOVE_X#
SetUnitData I&, "encDistY", CONVEYOR_MOVE_Y#

GetUnitData I&, "calcConvCalibPara", RESULT&

Elseif UnitItemIdent$(I&) = "" Then
    Exit Do
EndIf

I& = I& + 1

Loop While (1)

RESPONSECODE& = 0
RESPONSEVALUE&(0) = 0

EndIf
```

3-4-2 CPU ユニットのプログラム例

画像センサにキャリブレーションパラメータを送信するプログラム例を説明します。

ここでは、画像センサがキャリブレーションを行うシーンデータの作成は完了し、シーン切替されているものとします。

グローバル変数

名称	データ型	割付先 ^{*1}	コメント
E001_Line0_Command _Code	DWORD	ECAT://node#1/Line0 Command Code	コマンドコード
E001_Line0_Command _Parameter_0	DINT	ECAT://node#1/Line0 Command Parameter 0	コマンドパラメータ 0
E001_Line0_Command _Parameter_1	DINT	ECAT://node#1/Line0 Command Parameter 1	コマンドパラメータ 1
E001_Line0_Command _Parameter_2	DINT	ECAT://node#1/Line0 Command Parameter 2	コマンドパラメータ 2
E001_Line0_Command _Parameter_3	DINT	ECAT://node#1/Line0 Command Parameter 3	コマンドパラメータ 3
E001_Line0_Command _Ready	BOOL	ECAT://node#1/Line0 Command Ready	コマンド実行可能
E001_Line0_Command _Completion	BOOL	ECAT://node#1/Line0 Command Completion	コマンド実行完了
E001_Line0_Command _Request	BOOL	ECAT://node#1/Line0 Command Request	コマンド実行
E001_Line0_Command _Code_Echo_Back	DWORD	ECAT://node#1/Line0 Command Code Echo Back	コマンドコードエコーバック
E001_Line0_Trigger_Ready	BOOL	ECAT://node#1/Line0 Trigger Ready	トリガ入力可能状態
E001_Line0_Trigger	BOOL	ECAT://node#1/Line0 Trigger	計測トリガ
E001_Line0_Busy	BOOL	ECAT://node#1/Line0 Busy	処理実行中
E001_Line0_Trigger_Ack	BOOL	ECAT://node#1/Line0 Trigger Ack	トリガ受付状態
E001_Line0_User _Input_Area_4	LREAL	ECAT://node#1/Line0 User Input Area 4	ユーザ出力エリア 4
E001_Line0_User _Input_Area_5	LREAL	ECAT://node#1/Line0 User Input Area 5	ユーザ出力エリア 5
SimData	sSimData		キャリブレーションパラメータ

*1 画像センサをノードアドレス 1 に接続した場合の割付先です。

内部変数

名称	データ型	初期値	コメント
Trigger	BOOL	FALSE	実行条件
DoSetPara	BOOL	FALSE	処理中
Stage	INT	0	状態遷移

```

// Triggerの立ち上がりでシーケンス開始
IF Trigger AND NOT(DoSetPara) THEN
    DoSetPara:=TRUE;
    Stage:=1;
END_IF;

CASE Stage OF
1: // 画像モード変更
    E001_Line0_Command_Code:=DWORD#00000128;
    E001_Line0_Command_Parameter_0:=0;
    E001_Line0_Command_Parameter_1:=0;
    E001_Line0_Command_Parameter_2:=0;
    E001_Line0_Command_Parameter_3:=0;
    IF E001_Line0_Command_Ready AND NOT(E001_Line0_Command_Completion) THEN
        E001_Line0_Command_Request:=TRUE;
    END_IF;

    IF E001_Line0_Command_Completion THEN
        E001_Line0_Command_Request:=FALSE;
        IF E001_Line0_Command_Code_Echo_Back=E001_Line0_Command_Code THEN
            Stage:=2;
        END_IF;
    END_IF;

2: // 計測実行
    IF E001_Line0_Trigger_Ready THEN
        E001_Line0_Trigger:=TRUE;
    END_IF;
    IF E001_Line0_Trigger_Ack THEN
        E001_Line0_Trigger:=FALSE;
        Stage:=3;
    END_IF;

3: // キャリブレーション開始
    E001_Line0_Command_Parameter_0:=1;
    IF E001_Line0_Command_Ready AND NOT(E001_Line0_Command_Completion) THEN
        E001_Line0_Command_Request:=TRUE;
    END_IF;

    IF E001_Line0_Command_Completion THEN
        E001_Line0_Command_Request:=FALSE;
        IF E001_Line0_Command_Code_Echo_Back=E001_Line0_Command_Code THEN
            Stage:=4;
        END_IF;
    END_IF;

4: // MCS1を基準にした画像センサの原点座標の設定
    E001_Line0_Command_Parameter_0:=2;
    E001_Line0_User_Input_Area_4:=SimData.Vision[0].Origin[0];
    E001_Line0_User_Input_Area_5:=SimData.Vision[0].Origin[1];
    IF E001_Line0_Command_Ready AND NOT(E001_Line0_Command_Completion) THEN
        E001_Line0_Command_Request:=TRUE;
    END_IF;

    IF E001_Line0_Command_Completion THEN
        E001_Line0_Command_Request:=FALSE;
        IF E001_Line0_Command_Code_Echo_Back=E001_Line0_Command_Code THEN
            Stage:=5;
        END_IF;
    END_IF;

5: // MCS1を基準にしたコンベアの単位移動量の設定
    E001_Line0_Command_Parameter_0:=3;
    E001_Line0_User_Input_Area_4:=SimData.Vision[0].Dx;
    E001_Line0_User_Input_Area_5:=SimData.Vision[0].Dy;
    IF E001_Line0_Command_Ready AND NOT(E001_Line0_Command_Completion) THEN
        E001_Line0_Command_Request:=TRUE;
    END_IF;

    IF E001_Line0_Command_Completion THEN
        E001_Line0_Command_Request:=FALSE;
    
```

```

        IF E001_Line0_Command_Code_Echo_Back=E001_Line0_Command_Code THEN
            Stage:=6;
        END_IF;
    END_IF;

6:    // 画像センサの視野の大きさ(長さ&幅)の設定
    E001_Line0_Command_Parameter_0:=4;
    E001_Line0_User_Input_Area_4:=SimData.Vision[0].Length;
    E001_Line0_User_Input_Area_5:=SimData.Vision[0].Width;
    IF E001_Line0_Command_Ready AND NOT(E001_Line0_Command_Completion) THEN
        E001_Line0_Command_Request:=TRUE;
    END_IF;

    IF E001_Line0_Command_Completion THEN
        E001_Line0_Command_Request:=FALSE;
        IF E001_Line0_Command_Code_Echo_Back=E001_Line0_Command_Code THEN
            Stage:=7;
        END_IF;
    END_IF;

7:    // 画像センサの視野の傾きの設定
    E001_Line0_Command_Parameter_0:=5;
    E001_Line0_User_Input_Area_4:=SimData.Vision[0].TiltAngle/90;
    IF E001_Line0_User_Input_Area_4=-1 THEN
        E001_Line0_User_Input_Area_4:=3;
    END_IF;
    IF E001_Line0_Command_Ready AND NOT(E001_Line0_Command_Completion) THEN
        E001_Line0_Command_Request:=TRUE;
    END_IF;

    IF E001_Line0_Command_Completion THEN
        E001_Line0_Command_Request:=FALSE;
        IF E001_Line0_Command_Code_Echo_Back=E001_Line0_Command_Code THEN
            Stage:=8;
        END_IF;
    END_IF;

8:    // キャリブレーション実行
    E001_Line0_Command_Parameter_0:=6;
    IF E001_Line0_Command_Ready AND NOT(E001_Line0_Command_Completion) THEN
        E001_Line0_Command_Request:=TRUE;
    END_IF;

    IF E001_Line0_Command_Completion THEN
        E001_Line0_Command_Request:=FALSE;
        IF E001_Line0_Command_Code_Echo_Back=E001_Line0_Command_Code THEN
            Stage:=0;
        END_IF;
    END_IF;

0:
    IF NOT(Trigger) THEN
        DoSetPara:=FALSE;
    END_IF;

END_CASE;

```


4

座標系の説明

この章では、ビジョン&ロボット統合シミュレーションの対象である、ピック&プレースのアプリケーションにおける座標系について説明しています。

4-1	座標系の基本	4-2
4-2	統合シミュレーションの座標系	4-3
4-2-1	搬入コンベアの座標系	4-3
4-2-2	搬入コンベアとロボットの関係	4-3
4-2-3	搬入コンベアと画像センサのカメラ視野の関係	4-7
4-2-4	搬出コンベアの座標系	4-9
4-2-5	搬出コンベアの ON/OFF センサについて	4-9
4-2-6	搬出コンベアとロボットの関係	4-10

4-1 座標系の基本

ここでは、ピック&ブレースアプリケーションで使用する座標系のうち、一般的な座標系について説明します。

NJ Robotics CPU ユニットが使用する座標系

座標系	説明
軸座標系 (ACS)	各軸固有の回転座標系、または直動座標系のことです。 ジョイント座標系やリンク座標系と呼ぶこともあります。 各軸に必ずひとつ作られる座標系です。
機械座標系 (MCS)	ロボット固有の直交座標系です。 ロボットを動かすための目標位置は、機械座標系(MCS)の原点位置からみた TCP の位置で設定します。 ロボット 1 台(軸グループひとつ)につき、必ず 1 つ作られる座標系です。
ユーザ座標系 (UCS)	ユーザが任意に定義できる直交座標系です。 ユーザ座標系(UCS)の原点位置(Tx,Ty,Tz,Rx,Ry,Rz)は、機械座標系(MCS)原点からの位置と回転で設定します。回転は、姿勢とも呼びます。 ユーザ座標系(UCS)は、ロボット 1 台につき最大 16 個設定できます。
ツール座標系 (TCS)	使用するツールに応じて、ユーザが設定する直交座標系です。 ツール座標系(TCS)は、ツール TCP を基準とした座標系です。ツール座標系(TCS)はロボット 1 台につき最大 16 個設定でき、デフォルトツール座標系と合わせて最大 17 個から選択できます。

各座標系の詳細は、□「NJ シリーズ NJ Robotics CPU ユニット ユーザーズマニュアル (SBCA-421)」を参照してください。

画像センサが使用する座標系

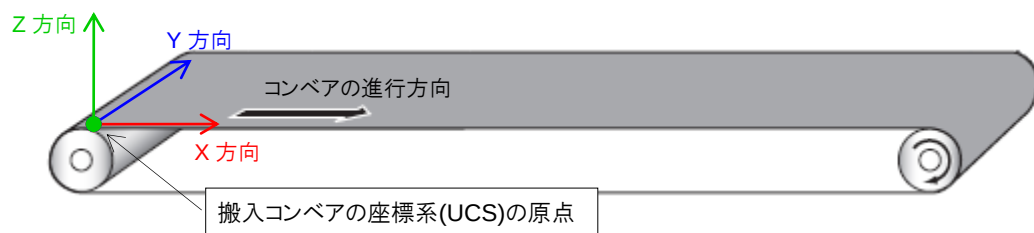
座標系	説明
カメラ座標	撮影した画像の左上を原点とし、水平方向を X 座標、垂直方向を Y 座標、単位が画素(ピクセル)のピクセル座標です。

4-2 統合シミュレーションの座標系

ここでは、ピック&プレースアプリケーションで使用する座標系のうち、統合シミュレーションが、キャリブレーションパラメータを算出するためや、ピック&プレースアプリケーションをシミュレーションするために使用している座標系について説明します。

4-2-1 搬入コンベアの座標系

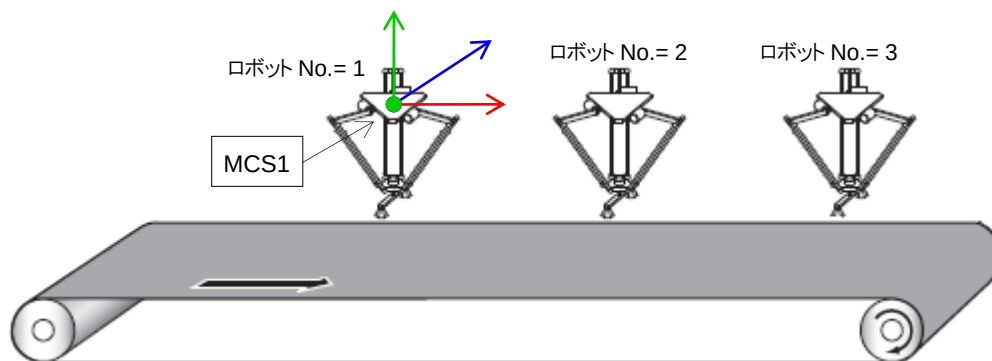
搬入コンベアの座標系には、ユーザ座標系(UCS)を使用しています。
搬入コンベアの上流端を、搬入コンベアの座標系(UCS)の原点としています。



統合シミュレーションが表示している搬入コンベア上のワークは、搬入コンベアの移動量(搬入コンベアに割り付けた軸の移動量)に合わせて、搬入コンベアの座標系(UCS)を X 方向に移動します。Y 方向、Z 方向には移動しません。

4-2-2 搬入コンベアとロボットの関係

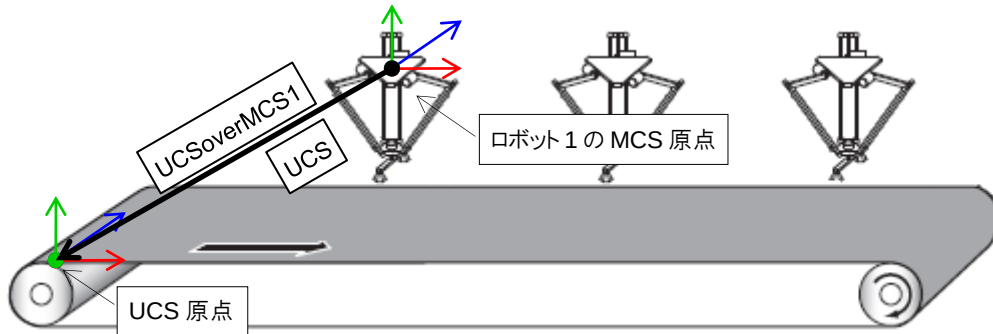
ロボット No.は、搬入コンベアに割り付いているロボットに対して、上流より 1 から昇順で割り付けます。
最上流に割り付いているロボットの機械座標系(MCS)を、MCS1 と呼びます。
これ以降、ロボット No.=x のロボットを、ロボット x と省略して呼びます。



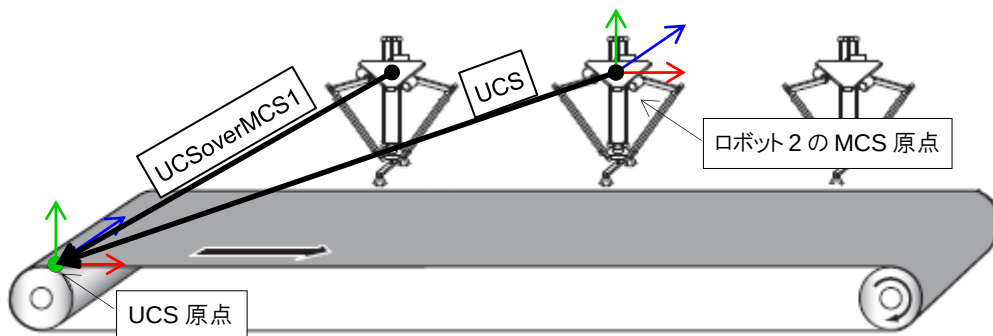
4-2 統合シミュレーションの座標系

搬入コンベアに複数台のロボットが割り付いている場合、搬入コンベアと各ロボットの位置関係は、ロボット1のUCS(MCS1を基準にしたUCS)と各ロボットのUCSから、算出することができます。ロボット毎のMCS1を基準にしたUCS(UCSoverMCS1)とUCSの位置関係は、以下のとおりです。

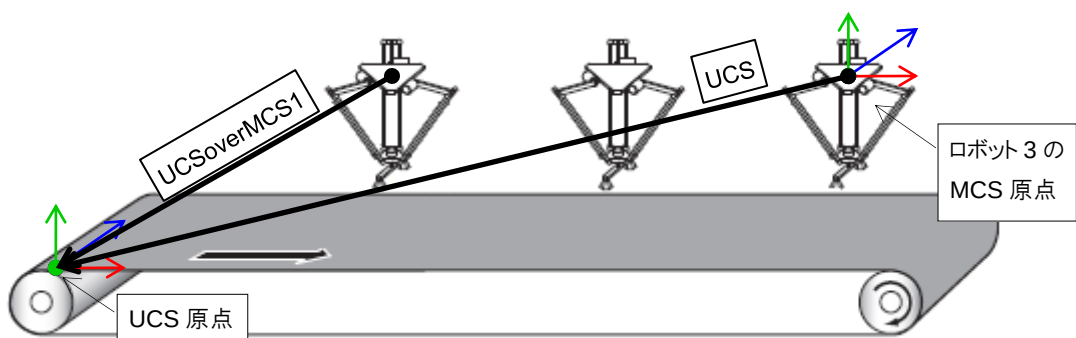
ロボット1のUCSoverMCS1とUCSの関係



ロボット2のUCSoverMCS1とUCSの関係



ロボット3のUCSoverMCS1とUCSの関係



ロボット1のUCSを設定するパラメータのうち、回転のパラメータ(Rx,Ry,Rz)が全てゼロの場合、搬入コンベア上のワークは、MCS1をX方向のみに移動し、MCS1におけるX方向の移動量は、搬入コンベアの移動量と一致します。

ロボット1のUCSを設定するパラメータのうち、回転のパラメータ(Rx,Ry,Rz)の一つ以上がゼロ以外の場合、搬入コンベア上のワークは、MCS1をX方向以外にも移動します。

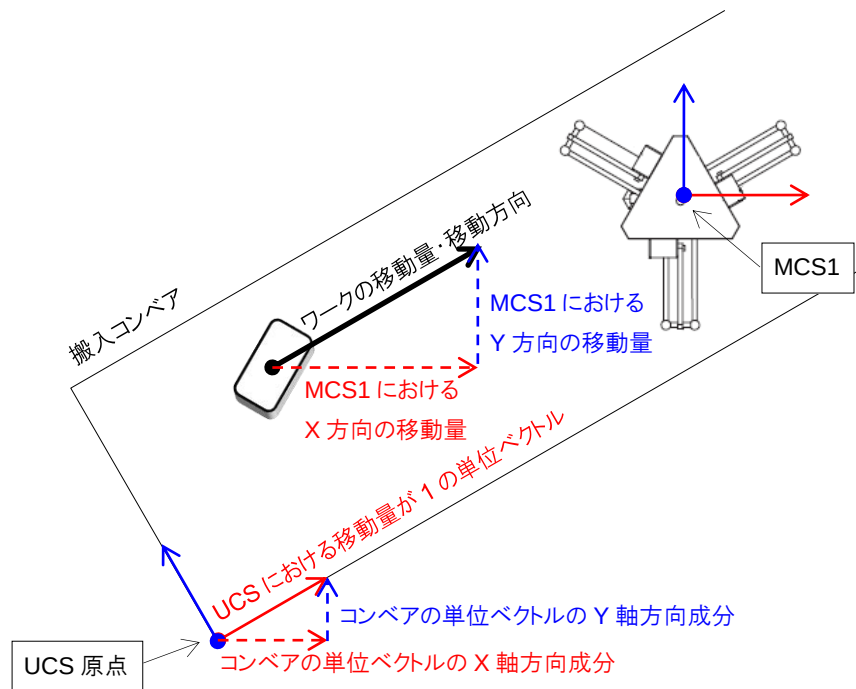
UCSの回転パラメータが全てゼロの場合、一つ以上がゼロ以外の場合、いずれの場合も、ロボット1の機械座標系(MCS1)を基準にした、搬入コンベア上のワークの移動量は、次式で表すことができます。

MCS1におけるX方向の移動量＝コンベアの移動量×コンベアの単位ベクトルのX軸方向成分

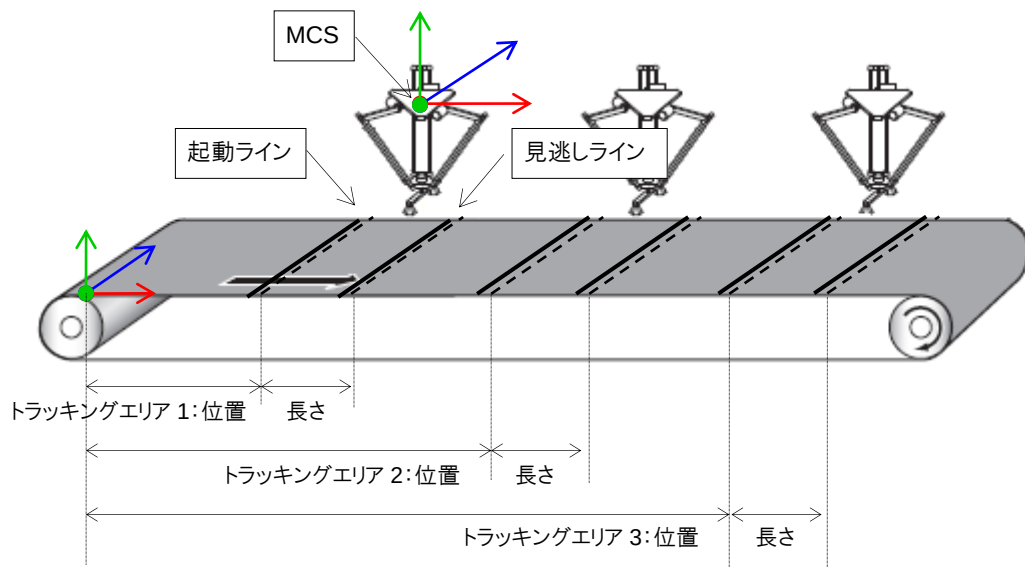
MCS1におけるY方向の移動量＝コンベアの移動量×コンベアの単位ベクトルのY軸方向成分

MCS1におけるZ方向の移動量＝コンベアの移動量×コンベアの単位ベクトルのZ軸方向成分

UCSのRzが30°の場合の、MCS1における、搬入コンベア上のワークの移動量



搬入コンベアに設定されたトラッキングエリアは、各ロボットの起動ライン、見逃しラインとして、ロボットの機械座標系(MCS)に変換されて、キャリブレーションパラメータとして出力されます。

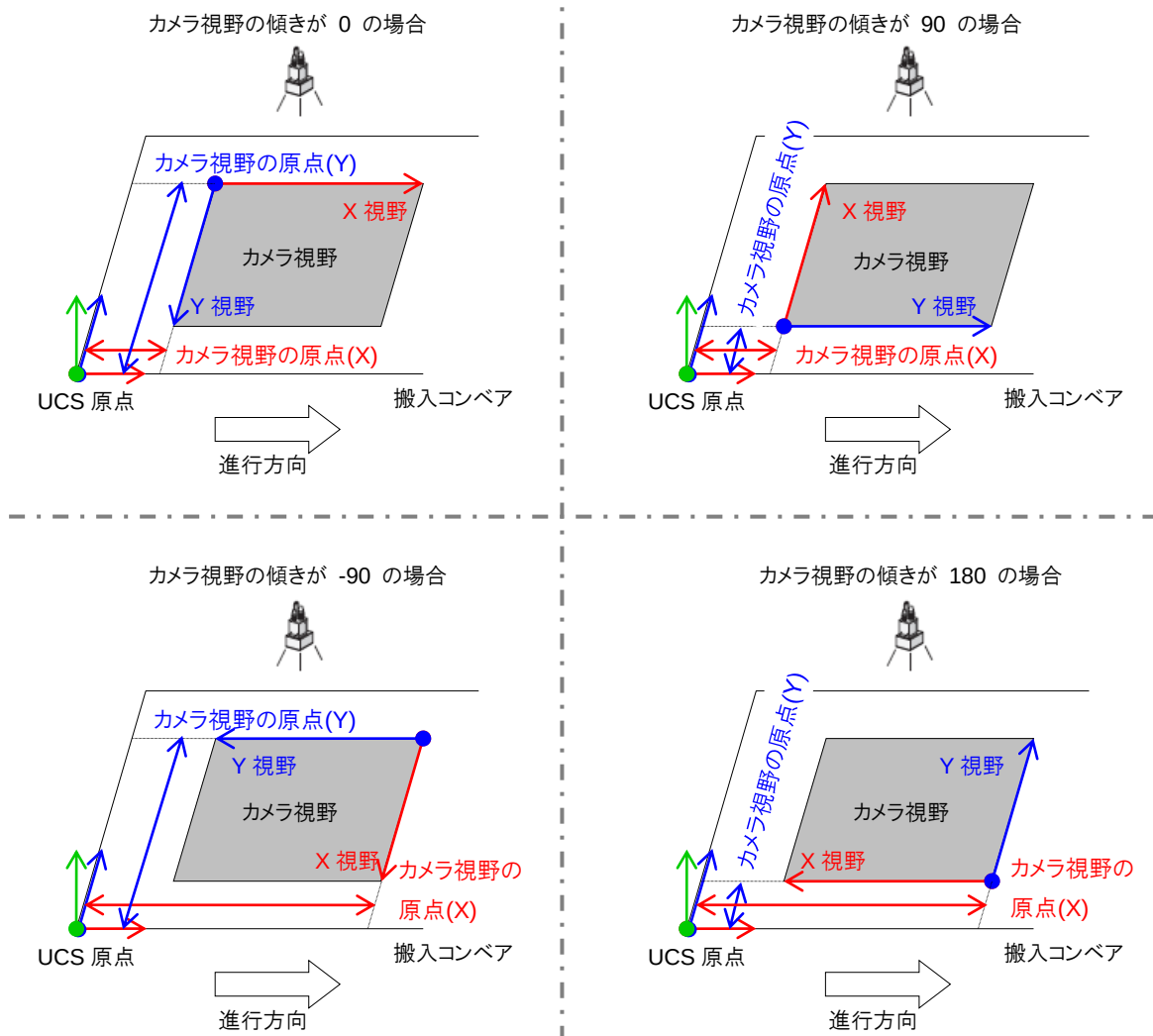


4-2-3 搬入コンベアと画像センサのカメラ視野の関係

画像センサのカメラ視野の座標(カメラ座標)は、搬入コンベアの座標系(UCS)のXY平面上に存在します。

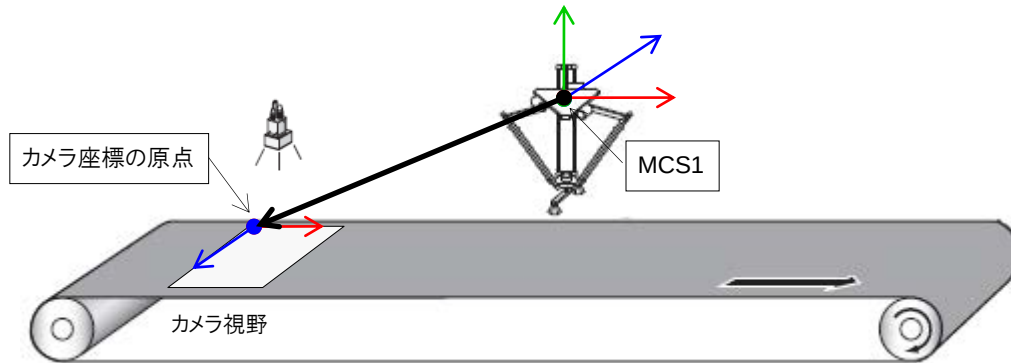
カメラ視野の傾きは、搬入コンベアの座標系(UCS)のX方向を基準に、カメラ視野の座標のX方向の傾き(回転角)を表しています。

カメラ視野の傾きに対する、搬入コンベアの座標系とカメラ座標の関係は、以下のとおりです。



4-2 統合シミュレーションの座標系

画像センサのカメラ視野の原点(カメラ座標の原点)は、ロボット1の機械座標系(MCS1)に変換されて、キャリブレーションパラメータとして出力されます。



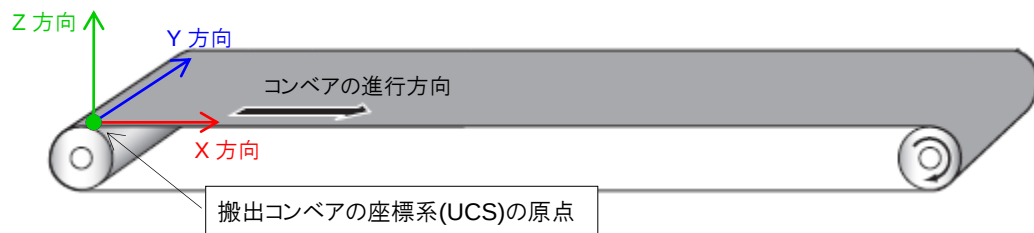
このキャリブレーションパラメータは、通信コマンドマクロを介して、画像センサにも渡してください。画像センサが出力するワークの位置は、このパラメータを使用して、ロボット1の機械座標系(MCS1)として出力するよう、シーンを作成してください。

統合シミュレーションは、画像センサが出力するワークの位置を、ロボット1の機械座標系(MCS1)として表示し、シミュレーションします。

ロボット1の機械座標系(MCS1)以外の座標系で、ワークの位置を出力した場合、統合シミュレーションは正しく表示できず、シミュレーションが行えません。

4-2-4 搬出コンベアの座標系

搬出コンベアの座標系には、搬入コンベア同様に、ユーザ座標系(UCS)を使用しています。搬出コンベアの上流端を、搬出コンベアの座標系(UCS)の原点としています。

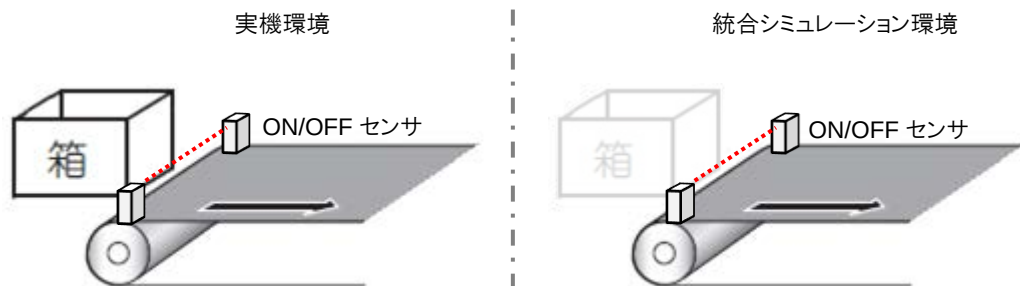


統合シミュレーションが表示している搬出コンベアの区切り、または、搬出コンベア上の箱は、搬出コンベアの移動量(搬出コンベアに割り付けた軸の移動量)に合わせて、搬出コンベアの座標系(UCS)をX方向に移動します。Y方向、Z方向には移動しません。

4-2-5 搬出コンベアの ON/OFF センサについて

搬出コンベアの座標系(UCS)の原点には、ON/OFF センサが存在します。実機環境において、搬出コンベア上の区切り、または、箱を検出するセンサを想定したものです。

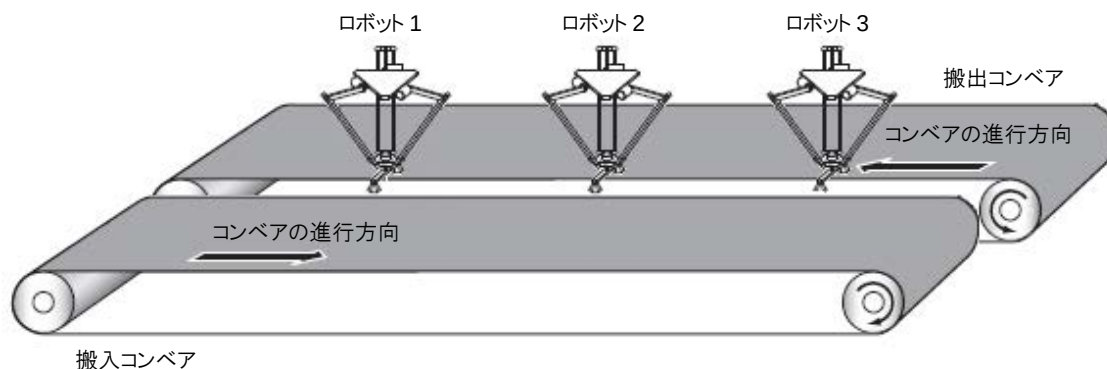
実機環境では、コンベアの区切りや、コンベア上の箱によって、ON/OFF されるセンサですが、統合シミュレーション環境では、ユーザプログラムから、ON/OFF を制御する必要があります。



統合シミュレーションは、搬出コンベアの ON/OFF センサが ON したタイミングで、コンベアに区切りを、あるいは、コンベア上に箱を表示し、シミュレーションします。

4-2-6 搬出コンベアとロボットの関係

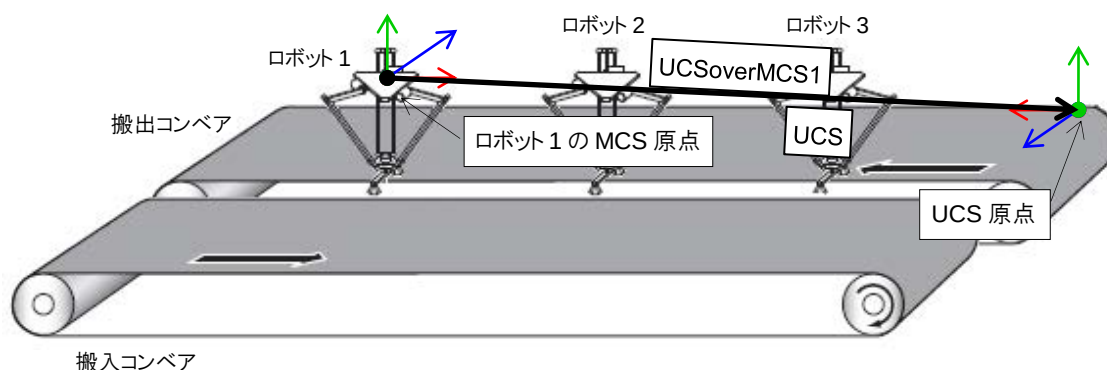
ロボット No.は、搬入コンベアを基準に割り付けるため、搬出コンベアのロボット No.は、上流より 1 から昇順で割り付くとは限りません。



上図のように、搬入コンベアと搬出コンベアが正対している場合、搬出コンベアのロボット No.は、上流より降順で割り付き、最下流がロボット 1 になります。

上図の場合、搬出コンベアの座標系(UCS)とロボット 1 の機械座標系(MCS1)の位置関係は、以下のとおりです。

ロボット 1 の UCSoverMCS1 と UCS



MEMO

MEMO

MEMO

●本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。
●本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物(又は技術)に該当するものを輸出(又は非居住者に提供)する場合は同法に基づく輸出許可、承認(又は役務取引許可)が必要です。

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

●製品に関するお問い合わせ先

お客様相談室

フリーダイヤル **0120-919-066**

携帯電話・PHS・IP電話などではご利用いただけませんので、下記の電話番号へおかけください。

電話 **055-982-5015** (通話料がかかります)

■営業時間：8:00～21:00 ■営業日：365日

●FAXやWebページでもお問い合わせいただけます。

FAX **055-982-5051** / www.fa.omron.co.jp

●その他のお問い合わせ

納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。

オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。

オムロン制御機器の最新情報をご覧ください。

www.fa.omron.co.jp

緊急時のご購入にもご利用ください。

オムロン商品のご用命は